Sparse-softmax: A Simpler and Faster Alternative Softmax Transformation

Shaoshi Sun[1], Zhenyuan Zhang[2], BoCheng Huang[3], Pengbin Lei[4], Jianlin Su[5], Shengfeng Pan[5], Jiarun Cao[6,*]

[1]School of Computer Science and Informatics, Cardiff University, the United Kingdom
[2]Department of Economics, Osaka City University, Japan
[3]School of Software Engineering, Beijing Jiaotong University, China
[4]College of Electronics and Information Engineering, Shenzhen University, China
[5] Shenzhen Zhuiyi Technology Co., Ltd., China
[6] Chinese Academy of Science, China

Abstract

The softmax function is widely used in artificial neural networks for the multiclass classification problems, where the softmax transformation enforces the output to be positive and sum to one, and the corresponding loss function allows to use maximum likelihood principle to optimize the model. However, softmax leaves a large margin for loss function to conduct optimizing operation when it comes to high-dimensional classification, which results in low-performance to some extent. In this paper, we provide an empirical study on a simple and concise softmax variant, namely sparse-softmax, to alleviate the problem that occurred in traditional softmax in terms of high-dimensional classification problems. We evaluate our approach in several interdisciplinary tasks, the experimental results show that sparse-softmax is simpler, faster, and produces better results than the baseline models.

The Softmax transformation is widely used in artificial neural networks for multi-class classification,multi-label classification and attention mechanisms where it typically appears as the last layer. However, when it comes to classification problems with high dimensional outputs(empirically more than 100 categories), the standard softmax and backpropagation do not take advantage of the sparsity of the categories and, as a result, softmax converges slowly on high-dimensional classification tasks. The softmax function has often been scrutinized in search of finding a better alternative to tackle the problem aforementioned. Specifically, the first direction is sampling methods approximations, which compute a fraction of the output's dimensions (Gutmann and Hyvärinen 2010; Mnih and Kavukcuoglu 2013; Mikolov et al. 2013; Shrivastava and Li 2014). The second direction is modeling high dimensional classification as a hierarchical classification task, where it modifies the output softmax layer by introducing heuristical-based tree (Mikolov et al. 2013; Morin and Bengio 2005).

Furthermore, (Vincent, De Brébisson, and Bouthillier 2015) explore the spherical loss family where they propose an alternative softmax that has log-softmax loss as one of its members. (de Brébisson and Vincent 2015) further work on this family of loss functions and propose log-Taylor softmax as a superior alternative than others, including original log-softmax loss. (Liu et al. 2016) propose large-margin softmax (LM-softmax) that tries to increase inter-class separation and decrease intra-class separation. This approach is further investigated by (Liang et al. 2017), where they propose soft-margin softmax (SM-softmax) that provides a ner control over the inter-class separation compared to LM-softmax.

In this work, we propose a simple and scalable alternative softmax namely Sparse-softmax, which specifically takes an effect on the high-dimensional classification problems. We first describe precisely our approach and explain theoretically why it is effective. Then we evaluate our approach on the interdisciplinary tasks including three NLP tasks: text classification (Yang et al. 2019; Howard and Ruder 2018), abstractive summarization (Al-Sabahi, Zuping, and Nadher 2018; Nallapati et al. 2016), question generation (Kundu and Ng 2018; Tay et al. 2018), as well as an image classification in Cifar-100 dataset. Experimental results indicate that our approach outperforms the baseline models. Our contribution can be summarized as the following:

- We introduce a simple softmax alternative called **sparse-softmax**, and its corresponding loss function during training.
- We explain the reason in-depth why **sparse-softmax** advances normal softmax in high-dimensional classification problem.
- We design interdisciplinary experiments to exhaustively analyse our model, where the experimental result verifies our model effectiveness.

Methods

In this section, we provide a brief overview of the softmax transformation and cross-entropy loss function in section . Then, We propose our sparse-softmax and a modified loss function in section , furthermore, we elaborate on the inner mechanism of the advance of our approach in high-dimensional classification problems.

Background

Definition. We denote the d-dimensional simplex by $\Delta^d = \{\boldsymbol{p} \in \mathbb{R}^d : \boldsymbol{p} \geqslant 0, \|\boldsymbol{p}\|_1 = 1\}$, where the set of vectors

*Corresponding author

represents probability distributions over d categories. Empirically, we consider a task as a high-dimensional classification problem if $d \geqslant 100$.

Softmax. We focus on the transformations that convert vectors in $\mathbb{R}^d$ to probability distributions in Δ^d. One of the most well-studied one is the **softmax** function that converts a vector of weights to a posterior label probabilities. The softmax function is defined as following:

$$p_i = Softmax(\boldsymbol{z})_i = \frac{e^{\boldsymbol{z}_i}}{\sum_{j=1}^{d} e^{\boldsymbol{z}_j}} \tag{1}$$

where the exponential function is executed on each element $\boldsymbol{z}_i$ of the input vector $\boldsymbol{z}$ and the output values are normalized by dividing by the sum of the entire exponentials. The normalization operation let each element in the output vector p_i sum up to 1.

Cross-entropy loss function. To derive the loss function for the softmax function we start out from the likelihood function that a given set of parameters of the model can result in the prediction of the correct class of each input sample, as in the derivation for the logistic loss function. The maximization of this likelihood can be written as:

$$\arg\max_{\theta} \mathcal{L}(\theta|\boldsymbol{t}, \boldsymbol{z}) \tag{2}$$

Maximizing this likelihood can also be done by minimizing the negative log-likelihood:

$$\mathcal{L}_{ce} = -\log p_t = \sum_i e^{\boldsymbol{z}_i} - \boldsymbol{z}_t \tag{3}$$

where t denotes target category and $\boldsymbol{z}$ is the logits deriving from the output of softmax layer.

Sparse-softmax Algorithm

Approach. A limitation of the conventional softmax function is that the resulting probability distribution always has full support across each $\boldsymbol{z}_i$, in another word, $softmax_i(\boldsymbol{z}) \neq 0$ for every $\boldsymbol{z}_i$. This is a weakness when it comes to high-dimensional classification problems where a sparse probability distribution is desired.

In this paper, we propose an alternative transformation, which we call sparse-softmax, to tackle the limitation aforementioned. The idea of sparse-softmax is intuitive and concise: we manually set up a hyperparameter k, then we only select the maximum k input values as a vector $\Omega_k \in \mathbb{R}^k$ to pass through the exponential normalized function, while others are masked as 0:

$$Sparse\,Softmax(\boldsymbol{z})_i = \begin{cases} \frac{e^{\boldsymbol{z}_i}}{\sum_{j \in \Omega_k} e^{\boldsymbol{z}_j}}, & \boldsymbol{z}_i \in \Omega_k, \\ 0, & \boldsymbol{z}_i \notin \Omega_k \end{cases} \tag{4}$$

where Ω_k is the set of top-k maximum indices of $\boldsymbol{z}_i$. Accordingly, the cross entropy loss function for sparse-softmax $\mathcal{L}_{sparse}$ is modified as:

$$\mathcal{L}_{sparse} = \log \sum_{i \in \Omega_k} e^{\boldsymbol{z}_i} - \boldsymbol{z}_t \tag{5}$$

where $\boldsymbol{z}_t$ is the logit of the targeted category.

Theoretical analysis In this part, we theoretically explain why our sparse-softmax is effective on high-dimensional classification tasks. As we elaborated in Equation 2 and 3, the conventional cross-entropy loss function can also be written as the following:

$$\mathcal{L}_{ce} = \log(1 + \sum_{i \neq t} e^{\boldsymbol{z}_i - \boldsymbol{z}_t}) \tag{6}$$

where t is the targeted category, $\boldsymbol{z}$ is the logits. Presumably, we classify the current sample correctly, that is, $\boldsymbol{z}_{max} = \boldsymbol{z}_t$. Therefore, we can derive the inequality:

$$\begin{aligned} \log(1 + \sum_{i \neq t} e^{\boldsymbol{z}_i - \boldsymbol{z}_{max}}) &\geq \log(1 + \sum_{i \neq t} e^{\boldsymbol{z}_{min} - \boldsymbol{z}_{max}}) \\ &= \log(1 + (n-1)e^{\boldsymbol{z}_{min} - \boldsymbol{z}_{max}}) \end{aligned} \tag{7}$$

where n denotes the number of categories. Then we set up a bound ϵ for cross entropy loss function. We are aware of the necessary condition for cross entropy to be less than or equal to ϵ is:

$$\log(1 + (n-1)e^{\boldsymbol{z}_{min} - \boldsymbol{z}_{max}}) \leq \epsilon \tag{8}$$

so we solve the equation 8:

$$\boldsymbol{z}_{min} - \boldsymbol{z}_{max} \geq \log(n-1) - \log(e^{\epsilon} - 1) \tag{9}$$

As an example:

$$\epsilon = \log 2 \approx 0.69 \tag{10}$$

In this case, we are aware that:

$$\log(e^{\epsilon} - 1) = 0 \tag{11}$$

therefore,

$$\boldsymbol{z}_{max} - \boldsymbol{z}_{min} \geq \log(n-1) \tag{12}$$

In another word, to make sore the cross entropy loss can be reduced to 0.69, the difference between maximum logit $\boldsymbol{z}_{max}$ and minimum logit $\boldsymbol{z}_{min}$ must be greater or equal to $\log(n-1)$. However, when it comes to high-dimensional classification problems where n is much greater, $\log(n-1)$ is a relatively massive but unnecessary margin for loss function.

Therefore, in terms of a classification problem, although we expect that logit of the targeted category is greater than any other non-targeted category, it can result in overfitting problem since conventional cross entropy loss tends to reduce this margin to a large extend, in which it makes the model overlearn the category distribution. However, the margin $\log(n-1)$ is relatively small in sparse-softmax as we diminish the number of category n to hyperparameter k, such that alleviate the overfitting problem caused by conventional softmax.

Experiments

In this section, we compare our sparse-softmax with conventional softmax on several tasks: text classification, abstractive summarization, question generation in the natural language processing field, as well as a image classification task in the

Table 1: Experimental results in text classification task

	WOS-46985		OOS-EVAL		RCV1-V2		IFLYTEK	
	MACRO F1	MICRO F1	MACRO F1	MICRO F1	MACRO F1	MICRO F1	MACRO F1	MICRO F1
SOFTMAX	82.15	82.65	95.53	95.19	62.04	80.52	43.29	59.5
SPARSE(K=1)	82.23	82.77	95.59	95.52	62.09	80.72	43.54	59.8
SPARSE(K=10)	82.71	82.95	95.86	95.45	62.64	81.40	43.37	59.1
SPARSE(K=20)	**83.31**	**83.50**	**96.08**	**95.88**	**63.17**	**81.52**	43.85	**60.7**
SPARSE(K=50)	82.77	82.98	95.66	95.32	62.21	81.16	43.97	60.2
SPARSE(K=100)	81.96	82.47	95.71	95.35	21.89	46.46	**44.25**	60.1

computer vision field. Our goal is not to achieve the state-of-the-art on each task but to observe the effect of replacing the original softmax with our sparse-softmax. In the subsection below, we will provide detailed experimental results on the downstream task , an efficiency analysis in section and the model performance under different settings of hyperparameter k in section .

Text Classification

Datasets. Text classification is the task of assigning an appropriate category to a given sentence or document. The categories depend on the chosen dataset and can range from topics. In our experiment, we chose 4 different datasets for evaluation, which are all beyond 100 categories. Among them, Web of Science(WOS-46985) dataset [1] contains 46,985 documents with 134 categories which include 7 parents categories. OOS-eval dataset [2] is the benchmark for evaluating the 150 types of user intents classification system in the presence of out-of-scope queries for the dialog system. Reuters Corpus Volume I (RCV1-v2) [3] consists of more than 800,000 news agency stories manually classified by Reuters Ltd for research purposes, each of which is assigned multiple topics. The total number of topics is 103. IFLYTEK [4] is a Chinese long text classification dataset, which contains more than 17,000 long text annotated data including various application topics related to daily life, where it has a total of 119 categories. The statistics of these datasets are presented in Table 2.

Experimental Settings. Among on these text classification datasets, except for the Chinese dataset iflytek that we use pre-trained model Nezha_base (Wei et al. 2019) as our baseline, we all use BERT_base (Devlin et al. 2018) as our baseline model on the other English datasets. According to the text length distribution of different datasets, as well as the maximum word length limited by Bert, we set the hyperparameters as shown in the following table 4. It is worth noting that since the text length of the WOS-46985 dataset is generally too long, we adopt the way of head+tail to truncate the information for the text beyond maximum length. Moreover, as the RCV1-V2 dataset contains a large scale of samples, we set epoch is 5 for accelerating the training process, while the rest of the datasets iterate over 20 epochs.

[1]https://data.mendeley.com/datasets/9rw3vkcfy4/6

[2]https://github.com/clinc/oos-eval

[3]https://archive.ics.uci.edu/ml/datasets/Reuter

[4]https://global.xfyun.cn/

Results. We compare the experimental results of both softmax and sparse-softmax with the setting of K = 1, 10, 20, 50, 100 in four datasets. Following (Johnson and Zhang 2016; Howard and Ruder 2018), we adopt macro F1 and mirco F1 score for evaluation and get the following results shown in Table 1. We can observe sparse-softmax all outperforms softmax in four text classification datasets. Empirically, when the number of categories is roughly 100 in certain datasets , the setting of hyperparameter K as 20 tends to achieve the best performance. It is worth mentioning that the model performance drops dramatically when we set k as 100 in RCV1-V2 dataset, since the number of categories in the RCV1-V2 dataset is 103 , we found that the model does not converge properly because of the gradient propagation of the remaining categories in the iterative process when the probability distribution is truncated.

Efficiency Analysis

The plots for training loss upon mini-batches for WoS-46985, OOS-eval, RCV1-v2 and IFLYTEK are given in Figure 1. It can be seen that compared with the traditional softmax, the loss function of sparse-softmax drops to a relatively low level while using fewer epochs when the parameters were consistent with softmax, which proves that sparse-softmax can converge faster in the high-dimensional classification tasks. Meanwhile, it may be pertinent to note that in Figure 1, we see fluctuation in the training loss for the softmax function, whereas the plot is comparatively smoother for sparse-softmax, which also indicates our approach can be less likely perturbed and more likely to converge.

Robustness to hyperparameter k

In this section, we show that our procedure is stable in its hyperparameter k. The theoretical results suggest that a wide range of k can give us statistical consistent guarantees of performance improvement against traditional softmax. Such robustness in hyperparameter is highly desirable since optimal tuning is not always feasible under certain circumstances, especially when no sufficient validation set or computational resources are available.

Empirically, when the number of categories is approximately 100 in the classification tasks , it achieves the state-of-the-art results when k is selected as 20. As shown in Figure 2, $k = 20$ achieves the best micro F1 scores, which are 60.7% and 96.1% in iflytek, oos_eval dataset respectively.

Table 2: Statistics of datasets in text classification task.

	WOS-46985	OOS-EVAL	RCV1-V2	IFLYTEK
TRAINING SET				
SAMPLES	32889	15100	775220	12133
CATEGORIES	134	150	103	119
AVERAGE LENGTH	99	8.3	120.2	289.0
MAXIMUM LENGTH	998	28.0	500.0	4282.0
DEVELOPMENT SET				
SAMPLES	9444	3100	21510	2599
CATEGORIES	134	150	103	119
AVERAGE LENGTH	200.3	8.3	120.1	289.8
MAXIMUM LENGTH	1262	24.0	499.0	1755
TEST SET				
SAMPLES	4652	5500	1191	2599
CATEGORIES	134	150	103	119
AVERAGE LENGTH	197.9	8.3	116.4	289.8
MAXIMUM LENGTH	691	25.0	499.0	1755

Table 3: Experimental results on SQuAD 1.1 dataset.

	BLEU-1	BLEU-2	BLEU-3	BLEU-4
BASELINE	51.49%	36.19%	27.33%	21.20%
SPARSE-SOFTMAX	**52.39%**	**36.81%**	**28.13%**	**22.02%**

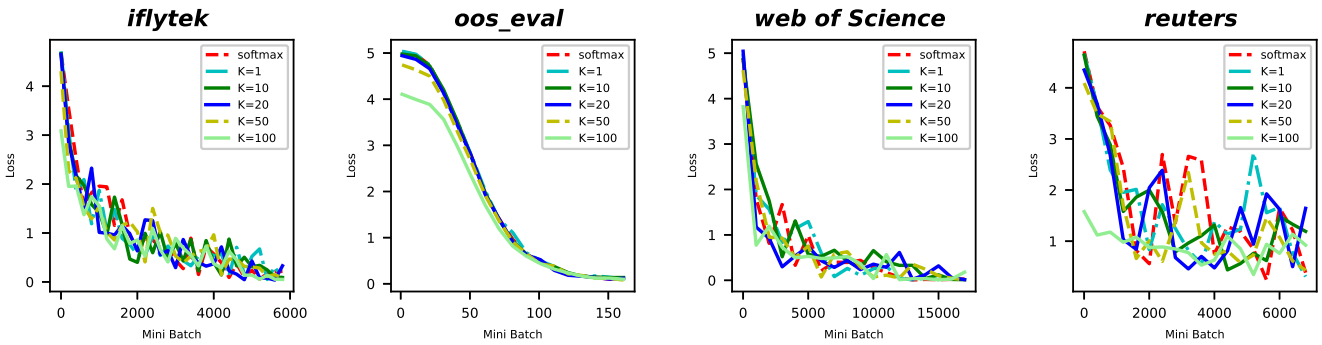

Figure 1: Loss curve in text classification task

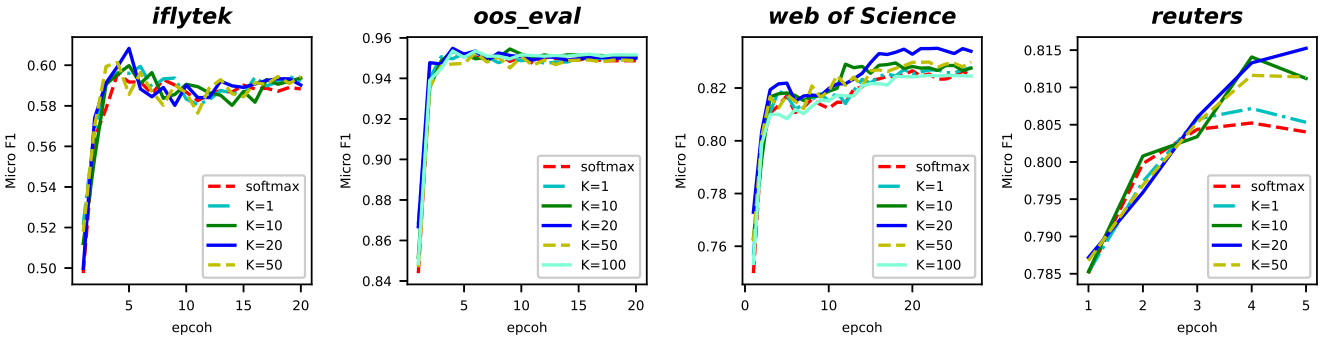

Figure 2: Micro F1 curve in text classification task

Auxiliary Experiments

To verify that sparse-softmax is adaptive across different high-dimensional tasks, we present experiments where we adopt sparse-softmax to the baseline models and compare the model performance respectively. The details are listed in the

Table 4: Experimental settings in text classification task

DATASET	MAX LENGTH	BATCH SIZE
WOS-46985	360	16
OOS-EVAL	64	256
RCV1-V2	512	16
IFLYTEK	256	24

following subsections including abstractive summarisation and question answering tasks in the natural language processing field, as well as image classification in the computer vision field.

Abstractive Summarisation

Automatic text summarization produces a concise and fluent summary conveying the key information in the input (e.g., a news article). We focus on abstractive summarization, a generation task where the summary is not constrained to reuse the phrases or sentences in the input text. We use Gigaword-10k [5] dataset for evaluation. We also use UNILM (Dong et al. 2019) as our baseline model and fine-tune the model on training set for 30 epochs. The masking probability is 0.7. We also use label smoothing (Müller, Kornblith, and Hinton 2019) with a rate of 0.1. For Gigaword, we set the batch size to 64, and maximum length to 256. During decoding, we experiment two different beam sizes: 1 and 5, respectively. We also fine-tune UNILM as a sequence-to-sequence model in our task. Due to the massive scale of the input vocabulary, the model needs to classify over 25,731 categories at each time step in the decoding stage if none of the post-processing steps are involved. Even if we mask all the tokens not appearing in the input sentence, the model still needs to classify over 256 categories as we set up the input maximum length to 256.

We use the F1 version of ROUGE of that in UNILM (Dong et al. 2019) as the evaluation metric for our dataset. In Table 5, we compare the model performance of raw baseline with adding our sparse-softmax, we can notice that sparse-softmax outperform baseline by 0.67% ROUGE-1 and 0.79% ROUGE-2 when it sets beam size = 1. In terms of beam size = 5, we mask the entire probability distributions except for top 5 ones, and our model also outperforms 0.28% ROUGE-1 and 0.55% ROUGE-2 respectively.

Table 5: Experimental results on Gigaword-10k dataset

beam size = 1	ROUGE-1	ROUGE-2
Unilm + softmax	32.23%	13.34%
Unilm + sparse-softmax	**32.90%**	**14.13%**

beam size = 5	ROUGE-1	ROUGE-2
Unilm + softmax	32.82%	13.93%
Unilm + sparse-softmax	**33.10%**	**14.48%**

[5]https://paperswithcode.com/sota/text-summarization-on-gigaword-10k

Question Answering

We also conduct experiments for the answer-aware question generation task (Chaplot et al. 2018; Lai et al. 2017). Given an input passage and an answer span, our goal is to generate a question that asks for the answer. We conduct evaluation on The SQuAD 1.1 dataset [6]. The question generation task is also formulated as a sequence-to-sequence problem which means the model needs to classify over the entire vocabulary as same as abstractive summarisation task above.

We also use UNILM as our baseline and fine-tune it on the training set for 10 epochs. We set the batch size to 32, masking probability to 0.7, and learning rate to 2e-5. The rate of label smoothing is 0.1. We set beam size = 1, which is aligned with the baseline model. During decoding, we truncate the input to 464 tokens by selecting a passage chunk that contains the answer. We use BLEU-4 as evaluation metrics (Dong et al. 2019). The result is showing in Table 3, by adding the sparse-softmax, our model outperforms the baseline model by 0.90%, 0.62%, 0.80%, 0.82% in terms of BLEU-1, BLEU-2, BLEU-3, BLEU-4 respectively.

Using pre-trained models

Pretrained model provides more informative initialized parameters to accelerate the convergence speed of the model on specific tasks. However, it also suffers more overfitting problem under the same circumstances as non-pretrained models. As elaborated in Section , we infer that sparse-softmax is capable of alleviating overlearned problem, since it can diminish the gap between maximum logit and minimum logit where it is used to measure the margin of cross entropy loss function. Intuitively, we assume applying sparse-softmax to pretrained model is more effective as the overlearned problem is more obvious in the pretrained models. To verify our speculation and the generalization capability of our approach, we conduct the experiment in an image classification task to verify this hypothesis. We adopt Cifar-100 [7] as our dataset and Densenet 201 (Huang et al. 2017) as our baseline models. We set epoch is 200, batch size is 62, and k is 20 for sparse-softmax in the training stage. The result is shown in Table 6, there was a slight performance degradation after using Densenet + Sparse-softmax.

As we assume that sparse-softmax is only effective under the framework of pre-trained models, we carry out the same experiment with InceptionV3 (Szegedy et al. 2016), which is pre-trained on imageNet dataset [8]. After deploy the pre-training framework, our proposed sparse-softmax shows better performance, which verify the hypothesis that sparse-softmax is more suitable for pre-trained model structures.

[6]https://datarepository.wolframcloud.com/resources/SQuAD-v1.1

[7]https://www.cs.toronto.edu/ kriz/cifar.html

[8]http://www.image-net.org/

Table 6: Experimental results on image classification tasks

MODEL	TOP1 ACC
DENSENET(RAW)	0.762
DENSENET(SPARSE-SOFTMAX)	0.737
INCEPTIONV3(FINE TUNED)	0.771
INCEPTIONV3(SPARSE-SOFTMAX)	0.778

Conclusion

In this paper, we propose sparse-softmax, which is an alternative of traditional softmax but achieves sparse probability distributions in the output. Experimental results on various tasks verifies that sparse-softmax can convex faster than conventional softmax and gain better model performance in high-dimensional classification tasks. Experiments on the image classification task suggest that our approach is adaptable to different domains under pre-trained model structure.

References

Al-Sabahi, K.; Zuping, Z.; and Nadher, M. 2018. A hierarchical structured self-attentive model for extractive document summarization (HSSAS). *IEEE Access*, 6: 24205–24212.

Chaplot, D. S.; Sathyendra, K. M.; Pasumarthi, R. K.; Rajagopal, D.; and Salakhutdinov, R. 2018. Gated-attention architectures for task-oriented language grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

de Brébisson, A.; and Vincent, P. 2015. An exploration of softmax alternatives belonging to the spherical loss family. *arXiv preprint arXiv:1511.05042*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dong, L.; Yang, N.; Wang, W.; Wei, F.; Liu, X.; Wang, Y.; Gao, J.; Zhou, M.; and Hon, H.-W. 2019. Unified Language Model Pre-training for Natural Language Understanding and Generation. In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*.

Gutmann, M.; and Hyvärinen, A. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 297–304.

Howard, J.; and Ruder, S. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708.

Johnson, R.; and Zhang, T. 2016. Supervised and semi-supervised text categorization using LSTM for region embeddings. In *International Conference on Machine Learning*, 526–534. PMLR.

Kundu, S.; and Ng, H. T. 2018. A question-focused multi-factor attention network for question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Lai, G.; Xie, Q.; Liu, H.; Yang, Y.; and Hovy, E. 2017. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*.

Liang, X.; Wang, X.; Lei, Z.; Liao, S.; and Li, S. Z. 2017. Soft-margin softmax for deep classification. In *International Conference on Neural Information Processing*, 413–421. Springer.

Liu, W.; Wen, Y.; Yu, Z.; and Yang, M. 2016. Large-margin softmax loss for convolutional neural networks. In *ICML*, volume 2, 7.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26: 3111–3119.

Mnih, A.; and Kavukcuoglu, K. 2013. Learning word embeddings efficiently with noise-contrastive estimation. *Advances in neural information processing systems*, 26: 2265–2273.

Morin, F.; and Bengio, Y. 2005. Hierarchical probabilistic neural network language model. In *Aistats*, volume 5, 246–252. Citeseer.

Müller, R.; Kornblith, S.; and Hinton, G. 2019. When does label smoothing help? *arXiv preprint arXiv:1906.02629*.

Nallapati, R.; Zhou, B.; Gulcehre, C.; Xiang, B.; et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Shrivastava, A.; and Li, P. 2014. Asymmetric LSH (ALSH) for sublinear time maximum inner product search (MIPS). *Advances in neural information processing systems*, 27: 2321–2329.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.

Tay, Y.; Luu, A. T.; Hui, S. C.; and Su, J. 2018. Densely connected attention propagation for reading comprehension. In *Advances in Neural Information Processing Systems*, 4906–4917.

Vincent, P.; De Brébisson, A.; and Bouthillier, X. 2015. Efficient exact gradient update for training deep networks with very large sparse targets. *Advances in Neural Information Processing Systems*, 28: 1108–1116.

Wei, J.; Ren, X.; Li, X.; Huang, W.; Liao, Y.; Wang, Y.; Lin, J.; Jiang, X.; Chen, X.; and Liu, Q. 2019. NEZHA: Neural contextualized representation for chinese language understanding. *arXiv preprint arXiv:1909.00204*.

Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R. R.; and Le, Q. V. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, 5753–5763.