Investigation of Large-Margin Softmax in Neural Language Modeling

Jingjing Huo[1], Yingbo Gao[1,2], Weiyue Wang[1,2], Ralf Schlüter[1,2], Hermann Ney[1,2]

[1]Human Language Technology and Pattern Recognition Group, Computer Science Department
RWTH Aachen University, 52074 Aachen, Germany, [2]AppTek GmbH, 52062 Aachen, Germany

`jingjing.huo@rwth-aachen.de, {ygao, wwang, schlueter, ney}@cs.rwth-aachen.de`

Abstract

To encourage intra-class compactness and inter-class separability among trainable feature vectors, large-margin softmax methods are developed and widely applied in the face recognition community. The introduction of the large-margin concept into the softmax is reported to have good properties such as enhanced discriminative power, less overfitting and well-defined geometric intuitions. Nowadays, language modeling is commonly approached with neural networks using softmax and cross entropy. In this work, we are curious to see if introducing large-margins to neural language models would improve the perplexity and consequently word error rate in automatic speech recognition. Specifically, we first implement and test various types of conventional margins following the previous works in face recognition. To address the distribution of natural language data, we then compare different strategies for word vector norm-scaling. After that, we apply the best norm-scaling setup in combination with various margins and conduct neural language models rescoring experiments in automatic speech recognition. We find that although perplexity is slightly deteriorated, neural language models with large-margin softmax can yield word error rate similar to that of the standard softmax baseline. Finally, expected margins are analyzed through visualization of word vectors, showing that the syntactic and semantic relationships are also preserved.

Index Terms: large-margin, softmax, neural language model, speech recognition

1. Introduction

The language model is an important component of automatic speech recognition (ASR) systems [1, 2, 3, 4], and perplexity (PPL) is known to be closely correlated with word error rate (WER) [5, 6]. Nowadays, state-of-the-art language models are commonly modeled using neural networks [7, 8, 9, 10]. The language model aims to learn the probability of word sequences, which are normally decomposed in an auto-regressive manner. To capture long contextual dependencies, the recurrent neural network (RNN) can be applied, which often uses the cross entropy training criterion along with softmax [8, 10, 11].

The idea of applying large-margin to the softmax layer is used to encourage intra-class compactness and inter-class separability among learned features. In the field of face recognition there exists a line of work [12, 13, 14, 15, 16, 17] that studies large-margin in the softmax layer, providing significant improvements in performance. Considering that the vectors in the projection matrix before the last softmax layer in neural language models (NLM) are essentially feature vectors of the words, which resemble the feature vectors of images in face recognition, we are thus curious to examine the performances of the aforementioned margins in NLM.

Large-margin in NLM is not an unfamiliar concept. In [18],

a global-level margin that discriminates sentences is introduced. In contrast, this paper focuses on the margin between atomic-level word vectors. We apply different types of large-margins from face recognition to NLM. Our initial experiments show that using the largin-margin softmax from face recognition out-of-the-box for NLM deteriorates the PPL dramatically. We assume that this is due to the fundamental differences between words and faces in their class distributions. It is important to note that unlike in face recognition, the posterior probability of words in NLM is highly unbalanced. Zipf's law [19] is a common approximation of word frequency versus word rank in natural languages. In [20], the authors observe that NLM learn word vectors whose norms are closely related to word frequencies. Therefore, we conduct a series of experiments to compare various norm-scaling techniques for the word vectors. In addition, we implement a heuristic to scale the norms of the context vectors. It turns out that one of the norm-scaling methods slightly improves the PPL. When it is used along with the margin techniques, comparable WER to the baseline is achieved. Finally, to figure out the effects of margin techniques in NLM, we visualize the word vectors and observe that word vectors trained with large-margin softmax exhibit expected behaviors and "stretch" the word vectors to more evenly populate the embedding space.

2. Related Work

The minimum distance of feature vectors to a decision boundary is called the margin. The large-margin concept plays an important role in conventional machine learning algorithms such as the support vector machine [21] and boosting algorithm [22]. Its core idea is to maximize the margin during training, in the hope that it leads to greater separability between classes during testing. [23, 24] study discriminative training on language models. The authors in [25] introduce large margin into gaussian mixture models for phonetic classification and recognition task (multiway classification). Later in [26], they show a framework to train large margin hidden markov models in the more general setting of sequential (as opposed to multiway) classification in ASR. It has also been well studied in image processing. A novel loss function is proposed in [27] to encourage large-margin in any set of layers of a deep network. In this work, we concentrate on the traditional margin methods that only focus on the output layer to see if it contributes to NLM.

The weights of the output layer are essentially feature vectors of each class (image features in face recognition or word embeddings in NLM). The scores (logits) of each sample are obtained using the dot product between the feature and the context vector. When using the cross entropy criterion along with softmax, the logits are used to calculate the loss. There exists a line of work in face recognition that modifies the loss function such that the scores of the true labels are reduced during

training.

In [12], the score for the ground truth class is manipulated by multiplying the angle between the ground truth feature vector and the context vector by a constant integer term m. It leads to a decline of that score, which ultimately leads to greater angular separability between learned feature vectors. This shares the similar idea with A-Softmax Loss (SphereFace) [28]. However, SphereFace normalizes the weights by L2-norms in advance so that the learned features are restricted to be on a hypersphere manifold, which is consistent with the widely used assumption in the field of manifold learning for face recognition that face images lie on or close to a low-dimensional manifold [29, 30].

Later, Hao et al. [14] propose a large-margin cosine loss (CosFace) using L2 normalization for both the feature vectors and the context vector, and subtracting a margin m from the cosine function output. CosFace also leads to a large-margin in the angular space. Subsequently, an additive angular margin loss (ArcFace) is presented in [15] that adds a margin term m to the angle instead of multiplying an integer term as in SphereFace. While these designs look similar, the authors claim that ArcFace has a better geometric attribute. Compared to SphereFace and CosFace, which are nonlinear in the angular space, ArcFace has a constant linear angular margin.

Word frequencies and vector norms are key concepts for this paper. Zipf's law [19] states that the frequency of words in a corpus of natural language is inversely related to the rank of words. Further in [20], the authors identify the relation between the norms of word vectors and their frequency. The result shows that the logarithm of word counts is a good approximation of word vector norms. This inspires us to examine various norm-scaling techniques for word vectors.

3. Methodology

Assuming the long short-term memory (LSTM) network [31] is used for language modeling, the target word posterior probability using softmax in the last output layer can be written as:

$$P(y|i) = \frac{\exp l(y,i)}{\exp l(y,i) + \sum_{y' \neq y} \exp l(y',i)} \quad (1)$$

where y denotes the target next word (dependency on i is dropped for simplicity), y' is a running index in the vocabulary, i is a running index of positions in data and $l(y,i)$ denotes the logit calculation. When using the inner-product, it can be written as:

$$l(y,i) = h_i^T W_y + b_y \quad (2)$$
$$= ||h_i|| \cdot ||W_y|| \cdot \cos \theta_{y,i} + b_y \quad (3)$$

where h_i is the context vector and the output of the LSTM layer(s), W_y is the embedding vector, $\theta_{y,i}$ is the angle between the two and b_y is the bias term.

Commonly, the softmax output is used together with the cross entropy training criterion:

$$L = -\sum_i \log P(y|i) \quad (4)$$

3.1. Conventional Margin

All of the three margins used in this paper only vary in the calculation of the logit of the ground-truth class y. For Cos-Face (l_{COS}) and ArcFace (l_{ARC}), the authors of the original paper claim that the normalization on features is necessary to encourage feature learning in their approach. Moreover, it is better to set the norm of the context vector as constant if the model is trained from scratch. Hence, they set $||W_y|| = 1$ and $||h_i|| = s$, where s is some predefined constant. In contrast to ArcFace and CosFace, L-Softmax (l_{LSM}) does not normalize anything in advance. The three margins from face recognition are formally defined as follows:

$$l_{\text{COS}}(y,i) = s \times (\cos(\theta_{y,i}) - m) \quad (5)$$
$$l_{\text{ARC}}(y,i) = s \times \cos(\theta_{y,i} + m) \quad (6)$$
$$l_{\text{LSM}}(y,i) = ||h_i|| \cdot ||W_y|| \cdot \varphi(\theta_{y,i}) \quad (7)$$

where $\varphi(\theta_{y,i})$ is designed as:

$$\varphi(\theta_{y,i}) = (-1)^k \cos(m\theta_{y,i}) - 2k \quad (8)$$

While margin m in l_{COS} and l_{ARC} are non-negative real numbers, it must be a positive integer in l_{LSM}. Using (8), the monotonicity of $\varphi(\theta_{y,i})$ with respect to m can be guaranteed. k in $\varphi(\theta_{y,i})$ is an integer in the range of $[0, m-1]$ and $\theta \in [\frac{k\pi}{m}, \frac{(k+1)\pi}{m}]$.

3.2. Margin with Norm-scaling

We explore different norm-scaling techniques for word vectors and context vectors, which differ only in how they alter the norms of vectors. $f(y)$ defines the modifications to word vectors and $g(i)$ defines the modifications to context vectors:

$$f(y) = \begin{cases} ||W_y||, & \text{no-mod} \\ ||W_{\text{argmax}\{c\}}||, & \text{uniform} \\ \log(\exp(||W_{\text{argmax}\{c\}}||) - v \times y), & \text{log-rank} \\ ||W_{\text{argmin}\{c\}}|| + u \times c_y, & \text{unigram} \\ \log(c_y), & \text{log-unigram} \end{cases} \quad (9)$$

$$g(i) = \begin{cases} ||h_i||, & \text{no-mod} \\ \max\{||h||\}, & \text{max-norm} \end{cases} \quad (10)$$

with v and u defined as:

$$v = \frac{\exp(||W_{\text{argmax}\{c\}}||) - \exp(||W_{\text{argmin}\{c\}}||)}{V} \quad (11)$$

$$u = \frac{||W_{\text{argmax}\{c\}}|| - ||W_{\text{argmin}\{c\}}||}{\text{argmax}\{c\}} \quad (12)$$

where c is the count of words and V is the vocabulary size.

For *uniform* we assume that all word vectors have the same norm, and in this case we use the norm of the word vector with the largest count among all words in the training corpus. For *log-rank*, we expect that the norm of the word vector is linear with respect to its index before a logarithmic operation (assuming words are sorted in descending order by their counts). For *unigram*, we use scaled and shifted word counts as new word vector norms after normalization. Note that for *uniform*, *log-rank* and *log-unigram*, $f(y)$ is dynamically updated in each update step during training. For *log-unigram*, we take the logarithm of word count directly as the norm of that word. On the other hand, the heuristic *max-norm* scales the norm of the i-th context vector h_i using the largest norm in the batch where the h_i appears.

Finally, combining norm-scaling and margin techniques, the logit calculation can be reformulated as:

$$l(y,i) = g(i)f(y)\phi(\theta_{y,i}) + b_y \quad (13)$$

where $g(i)$ is selected between $||h_i||$ and $\max\{||h||\}$, $f(y)$ is selected among the five norm-scaling functions and $\phi(\theta_{y',i}) = \cos(\theta_{y',i})$ for $y' \neq y$, otherwise

$$\phi(\theta_{y,i}) = \begin{cases} \cos(\theta_{y,i}) - m, & \text{COS} \\ \cos(\theta_{y,i} + m), & \text{ARC} \\ \varphi(\theta_{y,i}), & \text{LSM} \\ \cos(\theta_{y,i}), & \textit{no-margin} \end{cases} \quad (14)$$

For all experiments in the next section, we always keep the bias term b_y, as according to our early experiments, dropping it slightly degrades the performance across all setups.

4. Experiments

We use two datasets to compare the effects of the aforementioned techniques: Switchboard (SWB) and Quaero English. SWB is a relatively small dataset, with a vocabulary size of 30K and 25M training tokens. Quaero has a vocabulary size of 128K and 49M training tokens. We use two-layer LSTM language models with hidden state sizes of 1024 and 2048 for SWB and Quaero, respectively. For Quaero, we also apply the sampled softmax [32] method to speed up training. In the following experiments, we fix the model architecture and only alter the softmax layer.

4.1. Conventional Margin

Considering that large-margin works well in face recognition, to grasp the preliminary understanding of its effects on our task we apply the large-margin techniques described in Section 3.1 out-of-the-box for NLM. As shown in Table 1, for LSM, the norms of vectors are retained as defined in Equation 7, and setting m as one means that there is no modification of cosine similarity. Therefore, the first row of the table gives us the baseline of this margin. Moreover, as required in [12] that m must be an integer, the minimal step is to increase m by one. As can be seen, even setting m to two would dramatically worsen the performance, we do not further increase m in this experiment.

Table 1: *PPL on SWB using margins from face recognition out-of-the-box.*

Method	m	PPL
Baseline	n/a	53.7
LSM	1	53.5
	2	390.3
ARC	0	66.9
	0.001	68.0
	0.003	80.7
	0.01	106.1
	0.03	170.4
COS	0	66.9
	0.001	70.3
	0.003	77.7
	0.01	108.0
	0.03	382.1

For COS and ARC, the modifications on feature vector norms are defined in Equation 5 and Equation 6. The feature vector W_y and context vector h_i are firstly normalized and then a large scalar $s = 64$ is used for re-scaling. The results show a clear trend that the bigger the margin term gets, the worse the performance is. Even when disabling the margin, i.e. $m = 0$,

PPL is much higher than the baseline. As the only changes to the calculation in this case is the re-scaling of vector norms, this suggests that normalizing the word vectors and re-scaling them to have norms of 64 is too harsh for NLM in concern.

4.2. Margin with Norm-scaling

Considering the pattern discovered in [20], that the norms of word vectors approximate the logarithm of word counts, as well as the results of our preliminary experiments, we believe that norms of vectors play a nonnegligible role in NLM. Simply reducing them and re-scaling them using a large constant seems improper in our case. Hence, our next step aims to figure out which kind of norm-scaling is more suitable for NLM. Specifically, we vary the norm-scaling setup on $g(i)$ and $f(y)$ and examine the PPL of the corresponding models.

The top half of Table 2 depicts the performance of five different norm-scaling techniques of word vectors as defined in (9), where $g(i)$ uses *no-mod*. As seen, all of them slightly worsen the performance. The bottom half of the table reports the performance when $g(i)$ uses *max-norm*. We can see that only applying the heuristic on the context vector gives the best performance on SWB, and using *max-norm* and *log-unigram* together slightly improves the PPL on Quaero. We go on to apply the best norm-scaling setup in combination with $\phi(\theta_{y,i})$ variants for NLM.

Table 2: *PPL on SWB or Quaero using different norm-scaling techniques. $g(i)$ and $f(y)$ being no-mod corresponds to the standard softmax baseline.*

$g(i)$	$f(y)$	SWB	Quaero
no-mod	no-mod	53.7	105.8
	uniform	56.8	108.4
	log-rank	56.8	108.1
	unigram	56.0	108.4
	log-unigram	53.8	107.4
max-norm	no-mod	**52.9**	104.3
	uniform	56.4	107.2
	log-rank	57.4	109.6
	unigram	54.6	106.1
	log-unigram	53.1	**104.1**

Table 3: *PPL on SWB combining norm-scaling and large-margin softmax.*

$g(i)$	$f(y)$	m	ARC	COS
no-mod	no-mod	0.001	54.5	54.5
	log-unigram	0.001	55.3	55.3
max-norm	log-unigram	0.001	55.3	54.9
	no-mod	0.001	**54.2**	**54.1**
		0.003	55.5	55.9
		0.006	57.7	58.4
		0.010	60.0	60.9

Now that we have good norm-scaling setups for both the context vectors and the word vectors, the logical next step is to assess the performance of various margins in combination with them. We choose the four best combinations of $g(i)$ and $f(y)$ in Table 2, and conduct large-margin experiments. First, we use a very small margin term for all of them, i.e. $m = 0.001$. As can be seen in the first four rows in Table 3, they do not differ much in PPL and none of them improves over the

Table 4: *PPL on Quaero combining norm-scaling and large-margin softmax.*

$g(i)$	$f(y)$	m	ARC	COS
no-mod	no-mod	0.001	**111.2**	**111.6**
	log-unigram	0.001	114.7	114.2
max-norm	log-unigram	0.001	113.7	112.7
	no-mod	0.001	114.0	112.0

baseline on SWB. Furthermore, as shown in Table 4, all of them deteriorate the PPL on Quaero to a large degree. To further verify, we tune the margin term under our best norm-scaling setting. The results in the last five rows in the Table 3 clearly show that the performance gets worse as m increases.

Last but not the least, we conduct LSTM recurrent neural network rescoring experiments as shown in Table 5 to make a final verdict of the application of large-margin softmax in NLM. The baseline system is based on the hybrid hidden Markov model neural network[33]. It is interesting to find that although PPL is deteriorated, NLM with large-margin softmax can yield the same WER as the baseline.

Table 5: *PPL and WER on SWB using ARC or COS with $m = 0.001$ and best norm-scaling techniques.*

Metrics		baseline	ARC	COS
PPL		52.9	54.2	54.1
WER	Switchboard	13.7	13.7	13.7
	Callhome	7.1	7.1	7.1
	Average	10.4	10.4	10.4

5. Analysis

To analyze the effects of large-margin in NLM, in this section we visualize the word embeddings trained with large-margin softmax as well as the standard softmax. For visualization, the dimensionality of word vectors is reduced to two by first applying principal component analysis and then using t-distributed stochastic neighbor embedding [34].

Figure 1 shows the word vectors in polar coordinates. For COS and ARC, we use the large-margin softmax in bold in Table 3. The vectors are scaled and rotated to align the word "the" in all plots. The points in blue are the top 100 frequent words in SWB, which already account for around 65% of the total running words. As can be seen, these vectors of frequent words obtained by large-margin softmax approaches in (b) and (c) are more separable than those obtained by standard softmax in (a). In other words, the word vectors are further "stretched" to more evenly populate the embedding space.

To further investigate if the word embeddings obtained by large-margin softmax maintain the word relations in general. We visualize some word groups in the second column ((d), (e) and (f)) of Figure 1. The words in red color series are pairs that have semantic similarity while word groups in blue color series have syntactic relations. As seen, even though the large-margin softmax makes the angles among words larger, it can still preserve the semantic and syntactic relations. For instance, words that share a similar meaning ("auto" - "car") are well gathered and the angle between word "she" and "herself" is almost the same as the angle between word "he" and "himself" in (f).

(a) softmax, top 100   (d) softmax, word groups

(b) COS, top 100   (e) COS, word groups
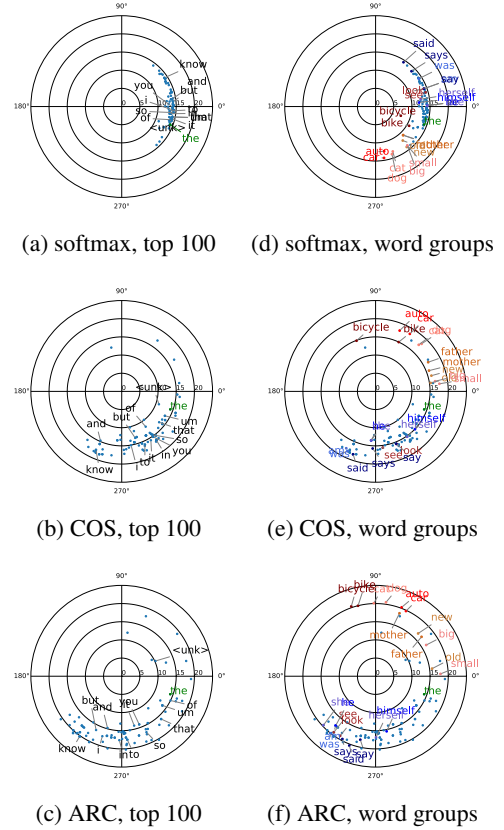
(c) ARC, top 100   (f) ARC, word groups

Figure 1: *Word vectors plotted in polar coordinates. In (a), (b) and (c), the top 100 frequent words more evenly populate the embedding space. In (d), (e) and (f), word groups with strong semantic (shades of red) or syntactic (shades of blue) relations are preserved.*

6. Conclusions

In this work, we investigate the use of large-margin softmax in neural language models. We first apply margins from face recognition out-of-the-box, which evidently deteriorates perplexity. Considering the unbalanced nature of word distributions, we further conduct experiments to find good norm-scaling settings for neural language models and tune the margin parameters. Then we apply the models trained with large-margin softmax in rescoring experiments, where we can reach the same word error rate performance as the standard softmax baseline. Finally, to figure out the effects of large-margin in neural language models, we visualize the word vectors. It is interesting to note that the expected margin are found among the word vectors trained with large-margin softmax, which makes them more evenly populate the embedding space. At the same time, the semantic and syntactic relations among words are also preserved.

7. Acknowledgements

8. References

[1] K. Irie, Z. Tüske, T. Alkhouli, R. Schlüter, and H. Ney, "LSTM, GRU, highway and a bit of attention: An empirical overview for language modeling in speech recognition," in *Interspeech*, San Francisco, CA, USA, Sep. 2016, pp. 3519–3523.

[2] X. Liu, Y. Wang, X. Chen, M. J. F. Gales, and P. C. Woodland, "Efficient lattice rescoring using recurrent neural network language models," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 4908–4912.

[3] Y. Si, Q. Zhang, T. Li, J. Pan, and Y. Yan, "Prefix tree based n-best list re-scoring for recurrent neural network language model used in speech recognition system," in *INTERSPEECH*, 2013.

[4] X. Liu, X. Chen, Y. Wang, M. J. F. Gales, and P. C. Woodland, "Two efficient lattice rescoring methods using recurrent neural network language models," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 8, pp. 1438–1449, Aug 2016.

[5] J. G. Wilpon, D. B. Roe *et al.*, *Voice communication between humans and machines*. National Academies Press, 1994.

[6] D. Klakow and J. Peters, "Testing the correlation of word error rate and perplexity," *Speech Commun.*, vol. 38, no. 1, pp. 19–28, Sep. 2002. [Online]. Available: http://dx.doi.org/10.1016/S0167-6393(01)00041-3

[7] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, Mar. 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=944919.944966

[8] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, "Recurrent neural network based language model," in *Eleventh annual conference of the international speech communication association*, 2010.

[9] H. Schwenk, "Continuous space language models," *Comput. Speech Lang.*, vol. 21, no. 3, pp. 492–518, Jul. 2007. [Online]. Available: http://dx.doi.org/10.1016/j.csl.2006.09.003

[10] M. Sundermeyer, H. Ney, and R. Schlüter, "From feedforward to recurrent LSTM neural networks for language modeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 517–529, Mar. 2015. [Online]. Available: http://dx.doi.org/10.1109/TASLP.2015.2400218

[11] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM neural networks for language modeling," in *Interspeech*, Portland, OR, USA, Sep. 2012, pp. 194–197.

[12] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ser. ICML'16. JMLR.org, 2016, pp. 507–516. [Online]. Available: http://dl.acm.org/citation.cfm?id=3045390.3045445

[13] X. Liang, X. Wang, Z. Lei, S. Liao, and S. Z. Li, "Soft-margin softmax for deep classification," in *ICONIP*, 2017.

[14] H. Wang, Y. Wang, Z. Zhou, X. Ji, Z. Li, D. Gong, J. Zhou, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," *CoRR*, vol. abs/1801.09414, 2018. [Online]. Available: http://arxiv.org/abs/1801.09414

[15] J. Deng, J. Guo, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," *CoRR*, vol. abs/1801.07698, 2018. [Online]. Available: http://arxiv.org/abs/1801.07698

[16] F. Wang, W. Liu, H. Liu, and J. Cheng, "Additive margin softmax for face verification," *CoRR*, vol. abs/1801.05599, 2018. [Online]. Available: http://arxiv.org/abs/1801.05599

[17] H. Liu, X. Zhu, Z. Lei, and S. Z. Li, "Adaptiveface: Adaptive margin and sampling for face recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[18] J. Huang, Y. Li, W. Ping, and L. Huang, "Large margin neural language model," *CoRR*, vol. abs/1808.08987, 2018. [Online]. Available: http://arxiv.org/abs/1808.08987

[19] D. M. Powers, "Applications and explanations of zipf's law," in *Proceedings of the joint conferences on new methods in language processing and computational natural language learning*. Association for Computational Linguistics, 1998, pp. 151–160.

[20] C. Herold, Y. Gao, and H. Ney, "Improving neural language models with weight norm initialization and regularization," in *Proceedings of the Third Conference on Machine Translation: Research Papers*. Belgium, Brussels: Association for Computational Linguistics, Oct. 2018, pp. 93–100. [Online]. Available: https://www.aclweb.org/anthology/W18-6310

[21] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep 1995. [Online]. Available: https://doi.org/10.1007/BF00994018

[22] R. E. Schapire, Y. Freund, P. Bartlett, W. S. Lee *et al.*, "Boosting the margin: A new explanation for the effectiveness of voting methods," *The annals of statistics*, vol. 26, no. 5, pp. 1651–1686, 1998.

[23] Z. Chen, K.-F. Lee, and M.-j. Li, "Discriminative training on language model," in *Sixth International Conference on Spoken Language Processing*, 2000.

[24] B. Roark, M. Saraclar, M. Collins, and M. Johnson, "Discriminative language modeling with conditional random fields and the perceptron algorithm," in *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2004, p. 47.

[25] F. Sha and L. K. Saul, "Large margin gaussian mixture modeling for phonetic classification and recognition," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 1. IEEE, 2006, pp. I–I.

[26] ——, "Large margin hidden markov models for automatic speech recognition," in *Advances in neural information processing systems*, 2007, pp. 1249–1256.

[27] G. Elsayed, D. Krishnan, H. Mobahi, K. Regan, and S. Bengio, "Large margin deep networks for classification," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 842–852. [Online]. Available: http://papers.nips.cc/paper/7364-large-margin-deep-networks-for-classification.pdf

[28] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," *CoRR*, vol. abs/1704.08063, 2017. [Online]. Available: http://arxiv.org/abs/1704.08063

[29] Y. Chang, C. Hu, and M. Turk, "Manifold of facial expression," in *Proceedings of the IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, ser. AMFG '03. Washington, DC, USA: IEEE Computer Society, 2003, pp. 28–. [Online]. Available: http://dl.acm.org/citation.cfm?id=946242.946277

[30] Xiaofei He, Shuicheng Yan, Yuxiao Hu, P. Niyogi, and Hong-Jiang Zhang, "Face recognition using laplacianfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 328–340, March 2005.

[31] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[32] S. Jean, K. Cho, R. Memisevic, and Y. Bengio, "On using very large target vocabulary for neural machine translation," *CoRR*, vol. abs/1412.2007, 2014. [Online]. Available: http://arxiv.org/abs/1412.2007

[33] M. Kitza, P. Golik, R. Schlüter, and H. Ney, "Cumulative adaptation for BLSTM acoustic models," in *Interspeech*, Graz, Austria, Sep. 2019, pp. 754–758.

[34] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.