CS222 Homework 3

Stable Matching and Algorithm Analysis

Exercises for Algorithm Design and Analysis by Li Jiang, 2018 Autumn Semester

- 潘佳萌

- 516030910510

1 low-rank matrix factorization

Tara N. Sainath has shown on three different LVCSR tasks ranging between 50-400 hrs, that a low-rank factorization reduces the number of parameters of the network by 30-50%. This results in roughly an equivalent reduction in training time, without a significant loss in final recognition accuracy, compared to a full-rank representation. And if the matrix is low-rank, we can use factorization to represent this matrix by two smaller matrices, thereby significantly reducing the number of parameters in the network before training. Another benefit of low-rank factorization for non-convex objective functions, such as those used in DNN training, is that it constrains the space of search directions that can be explored to maximize the objective function. This helps to make the optimization more efficient and reduce the number of training iterations, particularly for 2nd-order optimization techniques. This work extends these previous works by exploring low-rank factorization specifically for DNN training, which has the benefit of reducing the overall number of network parameters and improving training speed.

Their initial experiments are conducted on a 50-hour English Broadcast News (BN) task, where a DNN is trained with 2,220 output targets. they show by imposing a rank of 128 on the final matrix, they can reduce the number of parameters of the DNN by 28% with no loss in accuracy. Furthermore, they show that when low-rank matrices are used with 2nd order Hessian-free sequencetraining, they can reduce the overall number of training iterations by roughly 40%, leading to further training speed improvements. Second, they explore the behavior of low-rank factorization on two larger tasks with larger number of output targets: namely a 300-hour Switchboard (SWB) task with 9,300 output targets and a 400-hour English BN task with 6,000 output targets. On BN they can reduce the number of parameters of the network by 49% and for SWB by 32%, with nearly no loss in accuracy. Finally, they extend the use of low-rank factorization beyond acoustic modeling, exploring the versatility of the low-rank technique on DNNs used for language modeling (DNN-LM). they show that with low-rank factorization, we can reduce the number of parameters of a DNN-LM trained with 10,000 output targets by 45% without a significant loss in accuracy.

2 low-rank plus diagonal adaptation

The technique is inspired by observing that adaptation matrices are very close to an identity matrix or diagonally dominant. The LRPD restructures the adaptation matrix as a superposition of a diagonal matrix and a low-rank matrix. By varying the low-rank values, the LRPD contains the full and the diagonal adaptation matrix as its special cases.

Due to the high dimensionality of the DNN layers, the transformation matrix in a full form would result in a large speaker-dependent parameter set. When the amount of adaptation data is limited, overfitting becomes a severe issue. we decompose the adaptation matrix into two parts, one is a diagonal matrix and the other is a low-rank matrix which is further decomposed into two smaller matrices by SVD. This method can be applied to adapt either the full-size or the SVD SI DNN model. The latter leads to a combination of the SVD bottleneck and the LRPD adaptation.

Given an adaptation matrix, $W_{s,k*k}$ which can be $S_{s,k*k}$ in the SVD bottleneck adaptation, we approximate it as a superposition of a diagonal matrix $D_{s,k*k}$ and a low-rank matrix $L_{s,k*k}$ as

$W_{s,k*k} \simeq D_{s,k*k} + L_{s,k*k}$

The low-rank matrix $L_{s,k*k}$ can be represented as a product of two smaller matrices $P_{s,k*c}$ and $Q_{s,k*c}$. Hence

$W_{s,k*k} \simeq D_{s,k*k} + P_{s,k*c} + Q_{s,k*c}$

The number of elements in the LRPD decomposition is $k(2c + 1)$, while the originalWs has k2 elements. If $c \ll k$, this can significantly reduce the adaptation model footprint.

We can see that adaptation in an LRPD form amounts to inserting two linear layers above the layer being adapted. 1c. The first layer has c units with weight matrix Qs and the second one has k units with weight matrix Ps. The one-to-one skip-layer connections correspond to the diagonal component Ds.

Note that the LRPD adaptation can not only be applied to a full-size DNN, but also be applied to the bottleneck layer of a SVD DNN, leading to a combination of the SVD bottleneck adaptation and LRPD adaptation. This essentially forms a cascade of two low-rank models for different purposes. The SVD decompositionis aimed to restructure the SI DNN model, whereas the LRPD decomposition is to restructure the speaker-specific components.

The LRPD bridges the gap between the full and diagonal transformation matrices. When c = 0, the LRPD is reduced to adaptation with diagonal matrix. Specifically, if we apply the diagonal transforms before or after all non-linear layers, we may achieve the sigmoid or LHUC adaptation. On the other hand, if we fix the diagonal components Ds to an identity matrix, we achieve a form of low-rank plus identity (LRPI). Further setting c = 0 leads to the adaptation of only bias vectors.

参考文献

[1] T. N. Sainath, B. Kingsbury, B. Ramabhadran, P. Fousek,P. Novak, and A. Mohamed, “Making deep belief networks effective for large vocabulary continuous speech recognition,” in Proc. ASRU, 2011, pp. 30–35.

[2] Yong Zhao, Jinyu Li, and Yifan Gong,”low-rank plus diagonal adaptation for deep neural networks,” in Microsoft Corporation, One Microsoft Way, Redmond, WA 98052, USA,