SGD

A. Diagonal Hessian Matrix

For an intuitive explanation, we adopt the BN layer as an example to describe the design of SPH-DRS. For a BN layer, the 1D-parameters are formed by different values of parameters related to different channels. First we recall the operations of a single BN layer with C input channels. For a batch of input $X \in R^{N \times C \times W \times H}$ to this layer, divide it into C sets $X_1, \ldots, X_C$ according to the channels, where $X_i = \left\{x_i^1, x_i^2, \ldots, x_i^{m_i}\right\}$ for each channel index $i \in \{1, 2, \ldots, C\}$ and $m_i = NWH$ represents the number of the elements within the i-th channel. Then for a specific channel X_i, the BN layer contains the following operations:

$$\mu_{X_i} = \frac{1}{m_i} \sum_{j=1}^{m_i} x_i^j ; \sigma_{X_i}^2 = \frac{1}{m_i} \sum_{j=1}^{m_i} \left(x_i^j - \mu_{X_i}\right)^2 \tag{1}$$

$$\hat{x}_i^j = \frac{x_i^j - \mu_{X_i}}{\sqrt{\sigma_{X_i}^2 + \epsilon}} ; y_i^j = \gamma_i \hat{x}_i^j + \beta_i \tag{2}$$

The parameters γ_i, β_i introduced here are usually updated via back-propagation in the training process. Notice that the

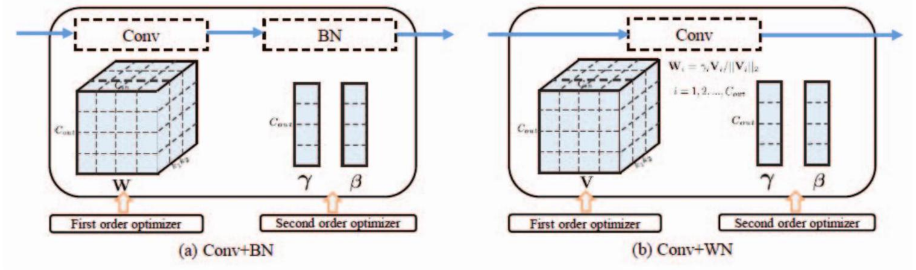

Fig. 1: Illustration of SPH-DRS. Here, Figure (a) represents the case when BN is adopted in neural networks training, while other normalization methods such as LN, GN and IN can be represented in the same way. Figure (b) illustrates the case of decoupling the convolutional layers when there are no normalization layers followed, where β represents bias, please see Section II-D for more details.

BN layer actually normalizes each channel independently, which means the parameters γ_i, β_i of X_i are irrelevant to γ_j, β_j of X_j whenever $j \neq i$. Thus, denote $\Gamma = (\gamma_1, \ldots, \gamma_C)^T$ as the group of parameters γ_i for all the C channels, the Hessian with respect to Γ, denoted by H_Γ here, is exactly a diagonal matrix. i.e.,

$$H_\Gamma := \frac{\partial^2 L}{\partial \Gamma^2} = \mathrm{Diag}\left(\frac{\partial^2 L}{\partial \gamma_1^2}, \ldots, \frac{\partial^2 L}{\partial \gamma_C^2},\right) \tag{3}$$

with $\frac{\partial^2 L}{\partial \gamma_i^2} = \sum_{j=1}^{m_i} \frac{\partial^2 L}{\partial \left(y_i^j\right)^2} \left(\hat{x}_i^j\right)^2$ for $i = 1, \ldots, C$. Then the inverse H_Γ^{-1}, if exists, can be computed directly by the inverse of each diagonal element. This structure states that getting the precise diagonal elements of the Hessian with respect to Γ is equivalent to obtaining the partial Hessian H_Γ precisely, which enables us to apply Newton-type methods easily. The same conclusion also holds for the Hessian of the group of parameters $\beta = (\beta_1, \ldots, \beta_C)^T$. Consequently, we can obtain the exact Hessian matrices of such one-dimensional parameters directly instead of using any approximation method.

It is worth mentioning that the property of partial diagonal Hessian does not hold for nature gradient descent methods (that is, the idea of extracting precise partial diagonal Hessian cannot be generalized to optimizers like KFAC), since the Fisher matrices related to 1D parameters are symmetric positive semidefinite matrices $E\left[gg^T\right]$ with g being the gradient vector, which cannot be proved to be diagonal.

B. The Hessian-free Approach

As we mentioned in Section I-A, the Hessian-free method can be regarded as a necessary way in designing second-order optimizers due to the memory limitations, and some existing optimizers have adopted Hessian-free methods to approximate the whole Hessian diagonal. Under this situation, it seems that there is no need to extract the diagonal of some specific layers. However, regarding the truth that the whole Hessian is not diagonal, the inexactness brought by approximating the whole diagonal may influence the optimization process,
which may lead to the unsatisfactory performance of the second-order optimizers on some tasks. As a consequence, by extracting the specific elements, we can avoid the inexactness of the existed diagonal approximation methods, which helps our optimization. In our optimizer, we adopt the Hessianfree approach with the help of back-propagation. To ensure continuity, we still take the BN layer as an example. Under the diagonal Hessian analysis Eq. 3, we set e_Γ to be the vector that takes elements 1 related to the 1 D vector Γ and takes elements 0 at all other places. Then with the help of the second time back-propagation, we can get

$$H_{e_\Gamma} = \frac{\partial g^T e_\Gamma}{\partial \Gamma} \tag{4}$$

where g is the gradient information computed in the first backpropagation process. Then the vector $\mathrm{diag}\left(H_\Gamma\right)$ is contained in the corresponding positions of the vector H_{e_Γ}. By this Hessian-free approach, we can get the corresponding diagonal matrix H_Γ precisely without computing and storing the whole Hessian matrix. Hence, our method avoids the inaccuracy brought by the iterative methods or the approximation methods when computing the Hessian information. The analysis of the group of bias parameters β in BN layers is the same as Γ. Apart from this, other normalization methods, including LN, IN and GN, also have analogous parameters, so similar derivations and conclusions can be obtained.

Generally, for an arbitrary 1D variable in DNN, we denote e_{SO} the vector takes 1 at positions corresponding to this 1 D vector and 0 at all other positions, and denote the partial diagonal Hessian matrix and the descent direction concerning it by H_{SO} and D_{SO}, respectively. Figure 2 illustrates the idea of our precise diagonal Hessian computation. Through this operation, we can obtain the precise partial Hessian for the calculation of descent direction.

C. Techniques in Nonconvex Optimization

Since a nonconvex objective function may not have a positive-definite Hessian, to apply the Newton method in

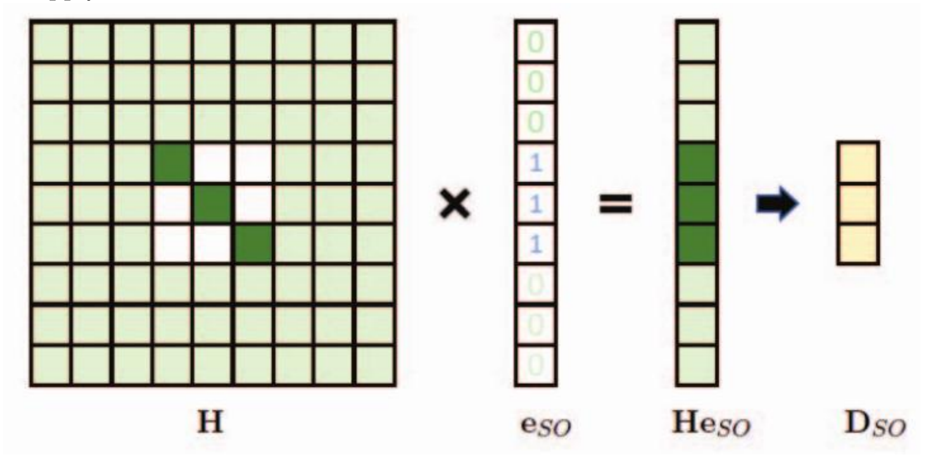

Fig. 2: Illustration of diagonal Hessian computation. Here, the light green boxes represent the elements that can be any real numbers while the white boxes represent zeros. The central 3×3 matrix (i.e., H_{SO}) in H is diagonal corresponding to the specific 1D variable. By multiplies with the vector e_{SO}, we can extract the diagonal elements precisely in the middle of He_{SO} and compute the element-wise inverse to get D_{SO}, which is exactly the diagonal of $H_S^{-1}O$.
solving nonconvex stochastic optimization problems, several rectification techniques are usually adopted to guarantee the well-definiteness and the effectiveness of the Newton method.

First, to handle the nonconvexity and noninvertible issues, we use the absolute value matrix with a positive perturbation as the substitution of the partial Hessian matrix, i.e.,