The Hessian of a Deep Net

Preconditioners like Adam, Shampoo, and RMS Prop are a common way to accelerte Stochastic Gradient Descent (SGD). These preconditioners approximately apply the inverse of the deep net's Hessian to the deep net's gradient. Such approximations are deemed necessary because the Hessian of a modern deep network is a very large matrix: A model that has a billion parameters, the Hessian has a quintillion elements, larger than what can be stored in a modern data center, let alone inverted.

This document shows that the Hessian of a deep net has a regular structure that makes it amenable to easy storage, multipcation and matrix inversion. Regardless of the operation of the layers, every twice differentiable deep net's Hessian is a quadratic in the inverse of a block-bi-diagonal matrix, and a number of block-diagonal matrices that depend on the differentials of the layers. This observation has two useful consequences: 1) It implies a fast way to multiply arbitrary vectors by the Hessian without storing the fully Hessian, 2) It implies a fast way to solve linear systems of equations in the Hessian without storing the Hessian or its inverse.

In an L-layer deep net where each layer has p parameters and produces a activations, naively storing the Hessian would require $O(L^2p^2)$ storage and just as many operations to multiply it by a vector, and naively solving a linear system in the the Hessian would require $O(L^3p^3)$ operations. By contrast, we show how to multiply a vector by the Hessian and solve linear systems in the Hessian with only $O(L\max(a,p)^2)$ operations.

Notation

We'll write a deep net as a pipeline of functions $\ell = 1, \ldots, L$,

$$z_1 = f_1(z_0; x_1) \tag{1}$$

$$\ldots \tag{2}$$

$$z_\ell = f_\ell(z_{\ell-1}; x_\ell) \tag{3}$$

$$\ldots \tag{4}$$

$$z_L = f_L(z_{L-1}; x_L) \tag{5}$$

The vectors $x_1, \ldots, x_L$ are the parameters of the pipeline. The vectors $z_1, \ldots, z_L$ are its intermediate activations, and z_0 is the input to the pipeline. The last layer f_L computes the final activations and their training loss, so the scalar z_L is the loss of the model on the input z_0. To make this loss's dependence on z_0 and the parameters explicit, we'll sometimes write it as $z_L(z_0; x)$. The foregoing formalization deviates slightly from the traditional deep net formalism in two ways: First, the training labels are subsumed in z_0, and are propagated through the layers until they're used in the loss. Second, the last layer fuses the loss (which has no parameters) and the last layer (which does).

Backpropagation and forward propagation, the recursive way

We would like to fit the vector of parameters $x = (x_1, \ldots, x_L)$ given a training dataset, which we'll represent by a stochastic input z_0 to the pipeline. Training the model proceeds by gradient descent steps along the stochastic gradient $\partial z_L(z_0; x)/\partial x$. The components of this direction can be computed by the chain rule with a backward recursion:

$$\frac{\partial z_L}{\partial x_\ell} = \underbrace{\frac{\partial z_L}{\partial z_\ell}}_{b_\ell} \underbrace{\frac{\partial z_\ell}{\partial x_\ell}}_{\nabla_x f_\ell}$$

$$\frac{\partial z_L}{\partial z_\ell} = \underbrace{\frac{\partial z_L}{\partial z_{\ell+1}}}_{b_{\ell+1}} \underbrace{\frac{\partial z_{\ell+1}}{\partial z_\ell}}_{\nabla_z f_{\ell+1}}$$

The identification $b_\ell \equiv \frac{\partial z_L}{\partial z_\ell}$, $\nabla_x f_\ell \equiv \frac{\partial z_\ell}{\partial x_\ell}$, and $\nabla_z f_\ell \equiv \frac{\partial z_\ell}{\partial z_{\ell-1}}$ turns this recurrence into

$$\begin{aligned} \frac{\partial z_L}{\partial x_\ell} &= b_\ell \cdot \nabla_x f_\ell \\ b_\ell &= b_{\ell+1} \cdot \nabla_z f_{\ell+1}, \end{aligned}$$

with the base case $b_L = 1$, a scalar.

Backpropagation, the matrix way

The above equations can be written in vector form as

$$\frac{\partial z_L}{\partial x} = \begin{bmatrix} \frac{\partial z_L}{\partial x_1} & \cdots & \frac{\partial z_L}{\partial x_L} \end{bmatrix} = \underbrace{\begin{bmatrix} b_1 & \cdots & b_L \end{bmatrix}}_{\equiv b} \underbrace{\begin{bmatrix} \nabla_x f_1 & & \\ & \ddots & \\ & & \nabla_x f_L \end{bmatrix}}_{\equiv D}$$

and

$$\begin{bmatrix} b_1 & b_2 & b_3 & \cdots & b_{L-1} & b_L \end{bmatrix} \underbrace{\begin{bmatrix} I & & & & \\ -\nabla_z f_2 & I & & & \\ & -\nabla_z f_3 & I & & \\ & & \ddots & \ddots & \\ & & & -\nabla_z f_L & 1 \end{bmatrix}}_{\equiv M} = \underbrace{\begin{bmatrix} 0 & \cdots & 1 \end{bmatrix}}_{\equiv e_L}.$$

Solving for b and substituting back gives

$$\frac{\partial z_L}{\partial x} = e_L M^{-1} D.$$

The matrix M is block bi-diagonal. Its diagonal entries are identity matrices, and its off-diagonal matrices are the gradient of the intermediate activations with respect to the layer's parameters. The matrix D is block diagonal, with the block as the derivative of each layer's activations with respect to its inputs. M is invertible because the spectrum of a triangular matrix can be read off its diagonal, which in this case is all ones.

The Hessian

The gradient we computed above is the unique vector v such that $dz_L \equiv z_L(x+dx) - z_L(dx) \to v(x)\cdot dx$ as $dx \to 0$. In this section, we'll compute the Hessian H of z_L with respect to the parameters. This is the unique matrix $H(x)$ such that $dv^\top \equiv v^\top(x+dx) - v^\top(x) \to H(x)\, dx$ as $dx \to 0$. We'll use the facts that $dM^{-1} = -M^{-1}(dM)M^{-1}$ and $b = e_L M^{-1}$ to write

$$
\begin{aligned}
dv &= d(e_L M^{-1} D) && (6)\\
&= e_L M^{-1}(dD) + e_L \left(dM^{-1}\right) D && (7)\\
&= b\cdot dD - e_L M^{-1}(dM)M^{-1}D && (8)\\
&= b\cdot dD - b\cdot (dM)M^{-1}D && (9)
\end{aligned}
$$

We'll compute each of these terms separately.

As part of this agenda, we'll need to define the gradient of tensor-valued functions $g:\mathbb{R}^d \to \mathbb{R}^{o_1\times\cdots\times o_k}$. We'll define this gradient $\nabla_x g(x) \in \mathbb{R}^{(o_1\cdots o_k)\times d}$ as the unique matrix-valued function that satisfies $\text{vec}\left(g(x+dx) - g(x)\right) \to \nabla_x g(x)\, dx$ as $dx \to 0$. This convention allows us to readily define, for example, the Hessian of a vector-valued function: If $g:\mathbb{R}^d \to \mathbb{R}^o$, then $\nabla_{xx} g(x) \in \mathbb{R}^{o\times d^2}$ is the unique matrix such that $\text{vec}\left(\nabla_x g(x+dx) - \nabla_x g(x)\right) \to \nabla_{xx} g(x)\, dx$. This convention also supports the chain rule as expected: For example, the gradient of $h(x) \equiv f(g(x))$ for a matrix-valued f and g can be written as $\nabla f \nabla g$ as expected. The chain rule in turn lets us define mixed partial derivatives, like $\nabla_{yz} g$ for $g:\mathbb{R}^{|x|}\to\mathbb{R}^{|g|}$. For example, if $y \in \mathbb{R}^{|y|}$ and $z \in \mathbb{R}^{|z|}$ are restrictions of $x \in \mathbb{R}^{|x|}$ to some $|y|$ and $|z|$-dimensional subsets, then $\nabla_z g(x) \in \mathbb{R}^{|g|\times|z|}$, and $\nabla_{yz} g(x) = \nabla_y \nabla_z g(x) \in \mathbb{R}^{|g||z|\times|y|}$. See Chapter 6 of Magnus & Neudecker for a deeper treatment of higher order derivatives of vector-valued functions.

The term involving dD

The matrix D is block-diagonal with its ℓth diagonal block containing the matrix $D_\ell \equiv \nabla_x f_\ell$. Using the facts that $\text{vec}(ABC) = \left(C^\top \otimes A\right)\text{vec}(B)$, and $(A\otimes B)^\top = A^\top \otimes B^\top$, we get

$$
\begin{aligned}
b\cdot(dD) &= \begin{bmatrix} b_1 & \cdots & b_L \end{bmatrix} \begin{bmatrix} dD_1 & & \\ & \ddots & \\ & & dD_L \end{bmatrix} && (10)\\
&= \begin{bmatrix} b_1\cdot dD_1 & \cdots & b_L\cdot dD_L \end{bmatrix} && (11)\\
&= \begin{bmatrix} \text{vec}\left(dD_1\right)^\top \left(I\otimes b_1^\top\right) & \cdots & \text{vec}\left(dD_L\right)^\top \left(I\otimes b_L^\top\right) \end{bmatrix} && (12)\\
&= \begin{bmatrix} \text{vec}\left(dD_1\right) \\ \vdots \\ \text{vec}\left(dD_L\right) \end{bmatrix}^\top \begin{bmatrix} I\otimes b_1^\top & & \\ & \ddots & \\ & & I\otimes b_L^\top \end{bmatrix} && (13)
\end{aligned}
$$

Observe that $\text{vec}\left(dD_\ell\right) = d\text{vec}\nabla_x f_\ell(z_{\ell-1}; x_\ell)$ varies with dx through both its arguments x_ℓ and $z_{\ell-1}$. Using mixed partials of vector-valued functions described above, we get

$$\text{vec}\left(dD_\ell\right) = d\text{vec}\left(\nabla_x f_\ell\right) = \left(\nabla_{xx} f_\ell\right)\ dx_\ell + \left(\nabla_{zx} f_\ell\right)\ dz_{\ell-1}.$$

Stacking these equations gives

$$\begin{bmatrix} \text{vec}\left(dD_1\right) \\ \vdots \\ \text{vec}\left(dD_L\right) \end{bmatrix} = \begin{bmatrix} \nabla_{xx} f_1 & & \\ & \ddots & \\ & & \nabla_{xx} f_L \end{bmatrix} dx + \begin{bmatrix} \nabla_{zx} f_1 & & \\ & \ddots & \\ & & \nabla_{zx} f_L \end{bmatrix} \begin{bmatrix} dz_0 \\ \vdots \\ dz_{L-1} \end{bmatrix}.$$

Each vector dz_ℓ in turn varies with dx via $dz_\ell = (\nabla_x f_\ell)\, dx_\ell + (\nabla_z f_\ell)\, dz_{\ell-1}$, with the base case $dz_0 = 0$, since the input z_0 does not vary with dx. Stacking up this recurrence gives

$$\begin{bmatrix} I & & & \\ -\nabla_z f_2 & I & & \\ & & \ddots & \\ & & -\nabla_z f_L & 1 \end{bmatrix} \begin{bmatrix} dz_1 \\ \vdots \\ dz_{L-1} \\ dz_L \end{bmatrix} = \begin{bmatrix} \nabla_x f_1 & & \\ & \ddots & \\ & & \nabla_x f_L \end{bmatrix} dx.$$

We can solve for the vector $\begin{bmatrix} dz_1 \\ \vdots \\ dz_L \end{bmatrix} = M^{-1} D\, dx$ and use the downshifting matrix

$$P \equiv \begin{bmatrix} 0 & & & \\ I & 0 & & \\ & \ddots & & \\ & I & 0 \end{bmatrix}$$

to plug back the vector $\begin{bmatrix} dz_0 \\ \vdots \\ dz_{L-1} \end{bmatrix} = PM^{-1} D\, dx$:

$$\begin{bmatrix} \operatorname{vec}(dD_1) \\ \vdots \\ \operatorname{vec}(dD_L) \end{bmatrix} = \left(\begin{bmatrix} \nabla_{xx} f_1 & & \\ & \ddots & \\ & & \nabla_{xx} f_L \end{bmatrix} + \begin{bmatrix} \nabla_{zx} f_1 & & \\ & \ddots & \\ & & \nabla_{zx} f_L \end{bmatrix} PM^{-1} D \right) dx.$$

The term involving dM

The matrix dM is lower-block-diagonal with $dM_2, \dots, dM_L$, and $dM_\ell \equiv d\nabla_z f_\ell$. Similar to the above, we can write

$$b \cdot (dM) M^{-1} D = \begin{bmatrix} b_1 & \cdots & b_{L-1} & b_L \end{bmatrix} \begin{bmatrix} 0 & & & \\ -dM_2 & 0 & & \\ & & \ddots & \\ & & -dM_L & 0 \end{bmatrix} M^{-1} D \tag{14}$$

$$= - \begin{bmatrix} b_2 \cdot dM_2 & \cdots & b_L \cdot dM_L & 0 \end{bmatrix} M^{-1} D \tag{15}$$

$$= - \begin{bmatrix} \operatorname{vec}(dM_2)^\top \left(I \otimes b_2^\top\right) & \cdots & \operatorname{vec}(dM_L)^\top \left(I \otimes b_L^\top\right) & 0 \end{bmatrix} M^{-1} D \tag{16}$$

$$= - \begin{bmatrix} \operatorname{vec}(dM_1) \\ \vdots \\ \operatorname{vec}(dM_L) \end{bmatrix}^\top \begin{bmatrix} 0 & & & \\ I \otimes b_2^\top & 0 & & \\ & & \ddots & \\ & & I \otimes b_L^\top & 0 \end{bmatrix} M^{-1} D \tag{17}$$

$$= - \begin{bmatrix} \operatorname{vec}(dM_1) \\ \vdots \\ \operatorname{vec}(dM_L) \end{bmatrix}^\top \begin{bmatrix} I \otimes b_1^\top & & \\ & \ddots & \\ & & I \otimes b_L^\top \end{bmatrix} PM^{-1} D. \tag{18}$$

Each matrix $dM_\ell = d\nabla_z f_\ell(z_{\ell-1}; x_\ell)$ varies with dx through both x_ℓ and $z_{\ell-1}$ as $d\operatorname{vec}(M_\ell) = (\nabla_{xz} f_\ell)\; dx_\ell + (\nabla_{zz} f_\ell)\; dz_{\ell-1}$. Following the steps of the previous section gives

$$\begin{bmatrix} \operatorname{vec}(dM_1) \\ \vdots \\ \operatorname{vec}(dM_L) \end{bmatrix} = \left(\begin{bmatrix} \nabla_{xz} f_1 & & \\ & \ddots & \\ & & \nabla_{xz} f_L \end{bmatrix} + \begin{bmatrix} \nabla_{zz} f_1 & & \\ & \ddots & \\ & & \nabla_{zz} f_L \end{bmatrix} PM^{-1} D \right) dx.$$

Putting it all together

We have just shown that the Hessian of the deep net has the form

$$H \equiv \frac{\partial^2 z_L}{\partial x^2} = D_D \left(D_{xx} + D_{zx} P M^{-1} D_x\right) + D_x^\top M^{-T} P^\top D_M \left(D_{xz} + D_{zz} P M^{-1} D_x\right)$$

The definitions below annotate the size of the various matrices in this expression assuming the first layer has a-dimensional activations ($z_1 \in \mathbb{R}^a$) and p-dimensional parameters ($x_1 \in \mathbb{R}^p$):

$$D_D \equiv \begin{bmatrix} \underbrace{I \otimes b_1}_{p \times ap} & & \\ & \ddots & \\ & & I \otimes b_L \end{bmatrix}, D_M \equiv \begin{bmatrix} \underbrace{I \otimes b_1}_{a \times a^2} & & \\ & \ddots & \\ & & I \otimes b_L \end{bmatrix}, P \equiv \begin{bmatrix} 0 & & & \\ I & 0 & & \\ & \ddots & \ddots & \\ & & I & 0 \end{bmatrix}$$

$$D_x \equiv \begin{bmatrix} \underbrace{\nabla_x f_1}_{a \times p} & & \\ & \ddots & \\ & & \nabla_x f_L \end{bmatrix}$$

$$D_{xx} \equiv \begin{bmatrix} \underbrace{\nabla_{xx} f_1}_{ap \times p} & & \\ & \ddots & \\ & & \nabla_{xx} f_L \end{bmatrix}, D_{xz} \equiv \begin{bmatrix} \underbrace{\nabla_{xz} f_1}_{a^2 \times p} & & \\ & \ddots & \\ & & \nabla_{xz} f_L \end{bmatrix},$$

$$D_{zx} \equiv \begin{bmatrix} \underbrace{\nabla_{zx} f_1}_{ap \times a} & & \\ & \ddots & \\ & & \nabla_{zx} f_L \end{bmatrix}, D_{zz} \equiv \begin{bmatrix} \underbrace{\nabla_{zz} f_1}_{a^2 \times a} & & \\ & \ddots & \\ & & \nabla_{zz} f_L \end{bmatrix},$$

$$M \equiv \begin{bmatrix} I & & & & \\ -\nabla_z f_2 & I & & & \\ & -\nabla_z f_3 & I & & \\ & & \ddots & \ddots & \\ & & & -\nabla_z f_L & 1 \end{bmatrix}.$$

The inverse of the Hessian

The above shows that the Hessian is a second order matrix polynomial in M^{-1}. While M itself is block-biadiagonal, M^{-1} is dense, so H is dense. Nevertheless, this polynomial can be lifted into a higher order object whose inverse is easy to compute:

$$\begin{aligned} H &= D_D D_{xx} + D_D D_{zx} P M^{-1} D_x + D_x^\top M^{-\top} P^\top D_M D_{xz} + D_x^\top M^{-\top} P^\top D_M D_{zz} P M^{-1} D_x \\ &= \begin{bmatrix} M^{-1} D_x \\ I \end{bmatrix}^\top \begin{bmatrix} P^\top D_M D_{zz} P & P^\top D_M D_{xz} \\ D_D D_{zx} P & D_D D_{xx} \end{bmatrix} \begin{bmatrix} M^{-1} D_x \\ I \end{bmatrix} \\ &= I + \begin{bmatrix} D_x \\ I \end{bmatrix}^\top \underbrace{\begin{bmatrix} M^{-\top} & \\ & I \end{bmatrix}}_{\hat{M}^{-\top}} \underbrace{\begin{bmatrix} P^\top D_M D_{zz} P & P^\top D_M D_{xz} \\ D_D D_{zx} P & D_D D_{xx} - I \end{bmatrix}}_{\equiv Q} \underbrace{\begin{bmatrix} M^{-1} & \\ & I \end{bmatrix}}_{\hat{M}^{-1}} \begin{bmatrix} D_x \\ I \end{bmatrix}. \end{aligned}$$

The Woodbury formula gives

$$H^{-1} = I - \begin{bmatrix} D_x \\ I \end{bmatrix}^\top \left(\left(\hat{M}^{-\top} Q \hat{M}^{-1} \right)^{-1} + \begin{bmatrix} D_x \\ I \end{bmatrix} \begin{bmatrix} D_x \\ I \end{bmatrix}^\top \right)^{-1} \begin{bmatrix} D_x \\ I \end{bmatrix} \tag{19}$$

$$= I - \begin{bmatrix} D_x \\ I \end{bmatrix}^\top \Bigg(\underbrace{\hat{M} Q^{-1} \hat{M}^\top + \begin{bmatrix} D_x D_x^\top & D_x \\ D_x^\top & I \end{bmatrix}}_{\equiv A} \Bigg)^{-1} \begin{bmatrix} D_x \\ I \end{bmatrix}. \tag{20}$$

The matrix Q^{-1} can be computed explicitly using the partitioned matrix inverse formula. Define the Schur complement $S = Q_{11} - Q_{12} Q_{22}^{-1} Q_{21}$, where Q_{ij} denote the i, jth block of Q as defined above. Then

$$Q^{-1} = \begin{bmatrix} S^{-1} & -S^{-1}Q_{12}Q_{22}^{-1} \\ -Q_{22}^{-1}Q_{21}S^{-1} & Q_{22}^{-1} + Q_{22}^{-1}Q_{21}S^{-1}Q_{12}Q_{22}^{-1} \end{bmatrix}.$$

The matrices Q_{11}, Q_{12}, Q_{21}, and Q_{22} are all block-diagonal. S is also block diagonal because Q_{11} and $Q_{12}Q_{22}^{-1}Q_{21}$ are both block-diagonal. Since all the terms involved in the blocks of Q^{-1} are block-diagonal, Q^{-1} has the same banded structure as Q.

The inverse of $A \equiv \hat{M}Q^{-1}\hat{M}^\top + \begin{bmatrix} D_x D_x^\top & D_x \\ D_x^\top & I \end{bmatrix}$ can be applied efficiently. Instead of applying the Woodbury formula again, we'll compute its $LDL^\top$ decomposition and apply the inverse of that decomposition. The $LDL^\top$ decomposition of A is

$$A = \begin{bmatrix} I & A_{12}A_{22}^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} A_{11} - A_{12}A_{22}^{-1}A_{12}^\top & 0 \\ 0 & A_{22} \end{bmatrix} \begin{bmatrix} I & A_{12}A_{22}^{-1} \\ 0 & I \end{bmatrix}^\top \tag{21}$$

$$A_{11} = M\left[Q^{-1}\right]_{11} M^\top + D_x D_x^\top \tag{22}$$

$$A_{12} = M\left[Q^{-1}\right]_{12} + D_x \tag{23}$$

$$A_{22} = \left[Q^{-1}\right]_{22} + I. \tag{24}$$

so

$$A^{-1} = \begin{bmatrix} I & -A_{12}A_{22}^{-1} \\ 0 & I \end{bmatrix}^\top \begin{bmatrix} A_{11} - A_{12}A_{22}^{-1}A_{12}^\top & 0 \\ 0 & A_{22} \end{bmatrix}^{-1} \begin{bmatrix} I & -A_{12}A_{22}^{-1} \\ 0 & I \end{bmatrix} \tag{25}$$

Since A_{11} is block tri-diagonal, A_{12} is block-bidiagonal, and A_{22} is block-diagonal, applying A^{-1} to a vector is fast.

To summarize, the following algorithm computes $H^{-1}g$ for an arbitrary vector g:

Algorithm: Computing $H^{-1}x$

Given a gradient vector $g \in \mathbb{R}^{Lp}$, computes $H^{-1}g$.

1. Compute the auxiliary vector

$$\begin{bmatrix} g_1' \\ g_2' \end{bmatrix} \in \mathbb{R}^{La+Lp} \equiv \begin{bmatrix} D_x \\ I \end{bmatrix} g.$$

D_x has L $a \times p$ blocks on its diagonal, so it takes Lap multiplications to compute v.

2. Form the banded matrix

$$A \in \mathbb{R}^{L(a+p)\times L(a+p)} \equiv \hat{M}Q^{-1}\hat{M}^\top + \begin{bmatrix} D_x D_x^\top & D_x \\ D_x^\top & I \end{bmatrix}.$$

To compute Q^{-1}, we first compute the blocks of Q. These take La^3 multiplications for $Q_{11} \in R^{La\times La}$, La^2p for $Q_{12} \in \mathbb{R}^{La\times Lp}$ and $Q_{21} \in \mathbb{R}^{Lp\times La}$, and La^2p for $Q_{22} \in \mathbb{R}^{Lp\times Lp}$. Computing $S \in \mathbb{R}^{La\times La}$ takes Lp^3 to compute Q_{22}^{-1}, and $2Lap^2$ to compute the product $Q_{12}Q_{22}^{-1}Q_{21}$. Given these quantities, for the blocks of Q^{-1}, it takes an additional La^3 to compute the upper left block, $L(a^2p + ap^2)$ to compute the off-diagonal blocks, and somewhat less than that to compute the bottom diagonal block since the matrices involved have already been computed. In all, it takes less than $9L\max(a,p)^3$ multiplications to compute Q^{-1}.

To compute $\hat{M}Q^{-1}\hat{M}^\top$ requires an additional $2La^3$ operations for a total of $11L\max(a,p)^3$ multiplications.

Finally computing and adding the second term requires La^2p multplications, bringing the tally to at most $12L\max(a,p)^3$ multiplications to compute A.

3. Apply A^{-1} to g':

$$\begin{bmatrix} g_1'' \\ g_2'' \end{bmatrix} = \begin{bmatrix} I & -A_{12}A_{22}^{-1} \\ 0 & I \end{bmatrix}^\top \begin{bmatrix} A_{11} - A_{12}A_{22}^{-1}A_{12}^\top & 0 \\ 0 & A_{22} \end{bmatrix}^{-1} \begin{bmatrix} I & -A_{12}A_{22}^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} g_1' \\ g_2' \end{bmatrix}$$

This computation requires $2L\max(a,p)^3$ multiplications to compute A_{22}^{-1} and $\left[A_{11} - A_{12}A_{22}^{-1}A_{12}\right]^{-1}$. The remaining operations are matrix multiplications that take at most $3L\max(a,p)^2$, which is smaller than Lp^3 when $p > 3$. This brings the tally to at most $15L\max(a,p)^3$ multiplications.

4. Compute the final result

$$y = g - \begin{bmatrix} D_x \\ I \end{bmatrix}^\top \begin{bmatrix} g_1'' \\ g_2'' \end{bmatrix}$$

These are against matrix-vector multipciations that take at most $L\max(a,p)^2$ when $p > 1$, bringing the tally to at most $16L\max(a,p)^3$.