Полное строгое определение локального и глобального Гессиана второго порядка в произвольной нейронной архитектуре

Аноним
Аффилиация
anonymous@example.com

June 9, 2025

Abstract

В данной статье представлено исчерпывающее математически строгое определение Гессиана второго порядка для нейронных сетей произвольной архитектуры, заданной направленным ациклическим графом. Существующие подходы к вычислению кривизны функции потерь нейронных сетей часто ограничиваются аппроксимацией Гаусса-Ньютона, учитывающей лишь часть вторых производных. В работе разработан полный формализм, учитывающий все чистые и смешанные вторые производные по входам и параметрам, кросс-блоки между разными параметрами, а также механизмы разделения параметров между узлами сети. Особое внимание уделено негладким активационным функциям через использование Clarke-Гессиана. Для тривиального графа из единственного узла без потомков и предков предложенные формулы сводятся к стандартному Гессиану $\nabla^2_\theta \mathcal{L}(\theta) \in \mathbb{R}^{p \times p}$. Предложенный формализм предоставляет теоретический фундамент для углубленного анализа геометрических свойств функционала потерь и разработки более эффективных алгоритмов оптимизации нейронных сетей произвольной архитектуры.

1 Введение

Гессиан второго порядка $\nabla^2 \mathcal{L}$ играет фундаментальную роль в анализе кривизны функционала потерь и в разработке методов оптимизации нейронных сетей [Martens, 2014, Pascanu et al., 2013b]. Методы второго порядка, такие как методы Ньютона, trust-region методы и их модификации, требуют точной информации о кривизне функции потерь для эффективной оптимизации [Nocedal and Wright, 2006]. Однако в контексте глубоких нейронных сетей вычисление и хранение полного Гессиана становится вычислительно неприемлемым, что приводит к необходимости использования различных аппроксимаций.

Наиболее распространенный подход — аппроксимация Гаусса-Ньютона, которая учитывает лишь часть всех вторых производных, игнорируя существенные компоненты кривизны [Schraudolph, 2002, Martens, 2010]. В данной работе мы предлагаем *полный* формализм, закрывающий следующие пробелы в существующей литературе:

- чистые и смешанные вторые производные по *входам* каждого узла нейронной сети;
- чистые вторые производные по *параметрам*;
- кросс-блоки $\partial^2/\partial\theta_v\,\partial\theta_w$ между параметрами разных узлов;
- смешанные вход-параметрические производные;
- учёт «разделения» (sharing) одного вектора параметров между несколькими узлами;
- обработка негладких активационных функций через методологию Clarke-Гессиана.

Особый случай: Если архитектура нейронной сети вырождается в единственный узел без потомков и предков, все предлагаемые определения естественным образом сводятся к стандартному Гессиану $\nabla^2_\theta \mathcal{L}(\theta) \in \mathbb{R}^{p\times p}$.

2 Связанные работы

Изучение геометрии функционала потерь нейронных сетей имеет долгую историю. Классические работы [Amari, 1998, Heskes, 2000] заложили основу для использования геометрической информации в оптимизации нейронных сетей. Особое значение имеет информационная геометрия и натуральный градиентный спуск, предложенный Амари [Amari, 1998].

В контексте вычисления Гессиана нейронных сетей, значительными являются работы [Martens, 2010, Martens and Sutskever, 2012], где представлены эффективные приближения Гессиана для глубоких нейронных сетей. Гаусс-Ньютон аппроксимация, которая игнорирует вторые производные функции потерь, часто применяется в практических алгоритмах из-за вычислительной эффективности и гарантии положительной полуопределенности.

Для негладких функций активации, таких как ReLU, традиционный анализ второго порядка неприменим. В работах [Clarke, 1990, Bolte and Pauwels, 2020] представлен обобщенный подход к недифференцируемым функциям через субдифференциальное исчисление и Clarke-градиенты. В нашей работе мы применяем эти концепции непосредственно к нейронным сетям, предлагая полный формализм для анализа кривизны функции потерь.

Недавние работы [Ghorbani et al., 2019, Sagun et al., 2017] исследуют спектральные свойства Гессиана функций потерь нейронных сетей и их связь с обобщением. Наше исследование дополняет эти работы, предоставляя точный математический аппарат для вычисления всех компонентов Гессиана в произвольных архитектурах нейронных сетей.

3 Методология

3.1 Таблица обозначений

Для удобства восприятия сложных формул и структур, приведём систематизированную таблицу основных обозначений:

Table 1: Основные обозначения, используемые в работе

Символ	**Определение**	**Размерность**
v, w, u	Узлы нейронной сети	—
$G = (V, E)$	Направленный ациклический граф, представляющий нейронную сеть	—
$\mathrm{Pa}(v)$	Множество родительских узлов узла v	—
$\mathrm{Ch}(v)$	Множество дочерних узлов узла v	—
f_v	Вектор выходов узла v	$\mathbb{R}^{d_v}$
θ_v	Вектор параметров узла v	$\mathbb{R}^{p_v}$
$\mathcal{L}$	Функция потерь	$\mathbb{R}$
δ_v	Градиент потерь по выходу узла v	$\mathbb{R}^{d_v}$
$D_{u\leftarrow v}$	Якобиан преобразования от узла v к узлу u	$\mathbb{R}^{d_u \times d_v}$
D_v	Якобиан выхода узла v по его параметрам	$\mathbb{R}^{d_v \times p_v}$
$T_{u;v}$	Тензор вторых производных выхода узла u по входу от узла v	$\mathbb{R}^{d_u \times d_v \times d_v}$
$T_{u;v,w}$	Тензор смешанных вторых производных по разным входам	$\mathbb{R}^{d_u \times d_v \times d_w}$
$T_{v;w,\theta}$	Тензор смешанных производных по входу и параметрам	$\mathbb{R}^{d_v \times d_w \times p_v}$
T_v^{θ}	Тензор вторых производных по параметрам	$\mathbb{R}^{d_v \times p_v \times p_v}$
$H_{v,w}^{f}$	Блок входного Гессиана между узлами v и w	$\mathbb{R}^{d_v \times d_w}$
H_{θ_v,θ_w}	Блок параметрического Гессиана	$\mathbb{R}^{p_v \times p_w}$
$\partial_C^2 f_v$	Clarke-Гессиан узла v (для негладкого случая)	множество матриц

Remark 1 (Соглашение об индексах). *В работе приняты следующие соглашения об индексах:*

- i *— индекс компоненты выхода узла (f_v или f_u)*
- j, k *— индексы компонент входов узлов*
- k, ℓ *— в контексте параметров, индексы компонент параметра* θ_v
- v, w, u *— индексы узлов в графе нейронной сети*

3.2 Функциональные пространства и аналитические предпосылки

Прежде чем перейти к определению компонентов и структуры Гессиана, необходимо формализовать функциональные пространства, в которых рассматривается задача, и уточнить аналитические предпосылки анализа.

Definition 1 (Функциональные пространства). *В рамках данной работы рассматриваются следующие функциональные пространства:*

- $\mathbb{R}^n$ *с евклидовой нормой* $\|\cdot\|_2$ *— конечномерное гильбертово пространство параметров, активаций и градиентов.*
- $C^2(\mathbb{R}^n, \mathbb{R}^m)$ *— пространство дважды непрерывно дифференцируемых функций из* $\mathbb{R}^n$ *в* $\mathbb{R}^m$*, используемое для гладкого случая.*
- $C^{1,1}(\mathbb{R}^n, \mathbb{R}^m)$ *— пространство непрерывно дифференцируемых функций с липшицевыми производными, используемое для негладкого случая.*
- $PC^2(\mathbb{R}^n, \mathbb{R}^m)$ *— пространство кусочно дважды дифференцируемых функций, где каждый кусок принадлежит* C^2*, а границы кусков образуют множество меры нуль.*
- $L(\mathbb{R}^n, \mathbb{R}^m)$ *— пространство линейных операторов (матриц) из* $\mathbb{R}^n$ *в* $\mathbb{R}^m$ *с операторной нормой и нормой Фробениуса.*

Assumption 1 (Регулярность функций узлов). *Для каждого узла* $v \in V$ *нейронной сети:*

1. *В гладком случае (Случай A): функция узла* $g_v \in C^2(\mathbb{R}^d, \mathbb{R}^{d_v})$*, т.е. дважды непрерывно дифференцируема по всем входам и параметрам.*
2. *В негладком случае (Случай B): функция узла* $g_v \in PC^2(\mathbb{R}^d, \mathbb{R}^{d_v}) \cap C^{1,1}(\mathbb{R}^d, \mathbb{R}^{d_v})$*, т.е. является кусочно дважды дифференцируемой с липшицевыми первыми производными (как, например, ReLU-активация), что обеспечивает существование и непустоту Clarke-субдифференциала.*

Proposition 1 (Существование и непустота Clarke-субдифференциала)**.** *Для локально липшицевой функции $f : \mathbb{R}^n \to \mathbb{R}$, которая дифференцируема почти всюду (в смысле меры Лебега), Clarke-субдифференциал $\partial_C f(x)$ определен и непуст во всех точках $x \in \mathbb{R}^n$. Более того, $\partial_C f(x)$ является выпуклым компактным множеством в метрическом пространстве $(L(\mathbb{R}^n, \mathbb{R}), \|\cdot\|_{op})$, где $\|\cdot\|_{op}$ — операторная норма.*

Для векторнозначных функций $F : \mathbb{R}^n \to \mathbb{R}^m$ субдифференциал определяется покомпонентно, и Clarke-Гессиан $\partial_C^2 F(x)$ также существует при соответствующих условиях локальной липшицевости и почти всюду дифференцируемости компонент градиента ∇F_i.

Эти предпосылки гарантируют корректность всех последующих определений и вычислений, связанных с дифференцированием функций узлов нейронной сети как в гладком, так и в негладком случаях.

3.3 Модель нейронной сети и обозначения

Definition 2 (Архитектура нейронной сети)**.** *Рассматривается нейронная сеть, архитектура которой представлена в виде направленного ациклического графа (DAG) $G = (V, E)$, где V — множество узлов сети, а E — множество направленных рёбер.*

Для каждого узла $v \in V$ определены следующие компоненты:

Входы: $f_{\mathrm{Pa}(v)} \in \prod_{u \in \mathrm{Pa}(v)} \mathbb{R}^{d_u}$, где $\mathrm{Pa}(v)$ — множество родительских узлов для v.

Параметры: $\theta_v \in \mathbb{R}^{p_v}$ — параметры, связанные с узлом v.

Функция узла: $f_v = g_v\big(f_{\mathrm{Pa}(v)}, \theta_v\big) \in \mathbb{R}^{d_v}$ — отображение, преобразующее входы и параметры в выход узла v.

Функция потерь: $\mathcal{L} : \mathbb{R}^{d_{out}} \to \mathbb{R}$ — функция потерь, определённая на выходном узле $out \in V$.

В зависимости от гладкости функций узлов, выделяем два принципиально различных случая:

- **Случай A (гладкий).** Все функции узлов g_v дважды непрерывно дифференцируемы по входам и параметрам, т.е. $g_v \in C^2$.
- **Случай B (негладкий).** В сети присутствуют негладкие функции активации, такие как ReLU, max-pooling и другие. В этом случае используется концепция Clarke-Гессиана $\partial_C^2 f_v$.

Вычисление всех блоков Гессиана осуществляется в *обратном топологическом порядке* по графу G, начиная с выходного узла *out*.

3.4 Градиенты первого порядка

Для разработки полного формализма Гессиана второго порядка необходимо сначала определить производные первого порядка, которые служат основой для дальнейших вычислений:

$$
\begin{aligned}
\delta_v &:= \nabla_{f_v} \mathcal{L} && \in \mathbb{R}^{d_v}, \quad \text{(градиент потерь по выходу узла } v\text{)} \\
\delta_{v,i} &:= [\delta_v]_i, && i = 1, \dots, d_v, \quad \text{(компоненты градиента)} \\
D_{u \leftarrow v} &:= \frac{\partial f_u}{\partial f_v} && \in \mathbb{R}^{d_u \times d_v}, \quad \text{(якобиан по входу)} \\
D_v &:= \frac{\partial f_v}{\partial \theta_v} && \in \mathbb{R}^{d_v \times p_v}. \quad \text{(якобиан по параметрам)}
\end{aligned}
$$

Градиенты δ_v и якобианы $D_{u \leftarrow v}$, D_v являются основой цепного правила первого порядка и используются для вычисления производных функции потерь по параметрам сети.

3.5 Тензоры вторых производных

Для полного учёта всех вторых производных функций узлов вводятся следующие тензорные структуры:

$$
\begin{aligned}
[T_{u;v}]_{i,j,k} &= \frac{\partial^2 (f_u)_i}{\partial (f_v)_j \, \partial (f_v)_k} && \in \mathbb{R}^{d_u \times d_v \times d_v}, && v \in \mathrm{Pa}(u), \\
[T_{u;\,v,w}]_{i,j,k} &= \frac{\partial^2 (f_u)_i}{\partial (f_v)_j \, \partial (f_w)_k} && \in \mathbb{R}^{d_u \times d_v \times d_w}, && v, w \in \mathrm{Pa}(u),\ v \neq w, \\
[T_{v;\,w,\theta}]_{i,j,k} &= \frac{\partial^2 (f_v)_i}{\partial (f_w)_j \, \partial (\theta_v)_k} && \in \mathbb{R}^{d_v \times d_w \times p_v}, && w \in \mathrm{Pa}(v), \\
[T_v^{\theta}]_{i,k,\ell} &= \frac{\partial^2 (f_v)_i}{\partial (\theta_v)_k \, \partial (\theta_v)_\ell} && \in \mathbb{R}^{d_v \times p_v \times p_v}.
\end{aligned}
$$

Remark 2 (Тензорная нотация и правила свертки)**.** *В тензорных выражениях выше и далее приняты следующие соглашения:*

- *Индекс i всегда относится к компоненте выхода соответствующего узла (f_u или f_v).*
- *Индексы j и k относятся к компонентам входов от родительских узлов.*
- *Индексы k и ℓ (в нижнем тензоре) относятся к компонентам параметров.*

- *Обозначение* $[T]_{i,\bullet,\bullet}$ *представляет матрицу (срез тензора), полученную фиксацией индекса* i.

- *При умножении на скаляр* $\delta_{u,i}$ *подразумевается свёртка по индексу* i *с весовыми коэффициентами* $\delta_{u,i}$.

При свертке тензоров с другими тензорами или векторами используются следующие правила:

- *Для выражения* $[T_{u;v}]_{i,j,k}\delta_{u,i}$ *результатом является матрица размерности* $d_v \times d_v$ *с элементами* $\sum_{i=1}^{d_u}[T_{u;v}]_{i,j,k}\delta_{u,i}$.

- *При матричном умножении* $D_{u\leftarrow v}^{\top}H_{u,u}^{f}D_{u\leftarrow w}$ *индексы сворачиваются согласно правилам матричного произведения, где* $D_{u\leftarrow v}^{\top} \in \mathbb{R}^{d_v\times d_u}$, $H_{u,u}^{f} \in \mathbb{R}^{d_u\times d_u}$, $D_{u\leftarrow w} \in \mathbb{R}^{d_u\times d_w}$.

- *Тензорное выражение* $\sum_{i=1}^{d_u}[T_{u;v,w}]_{i,\bullet,\bullet}\delta_{u,i}$ *преобразуется в матрицу размерности* $d_v \times d_w$ *с элементами* $\sum_{i=1}^{d_u}[T_{u;v,w}]_{i,j,k}\delta_{u,i}$.

Это соглашение обеспечивает однозначность всех тензорных операций в формулах и устраняет возможные неоднозначности при переходе от тензорной к матричной записи.

Эти тензоры учитывают чистые и смешанные вторые производные функций узлов по входам и параметрам. При суммировании по индексу i с весом $\delta_{u,i}$, эти тензоры дают вклад в Гессиан функции потерь.

3.6 Clarke-Гессиан для негладких функций активации

Для негладких функций активации, таких как ReLU, Leaky ReLU или max-pooling, классическое понятие Гессиана неприменимо в точках негладкости. В этом случае используется концепция Clarke-субдифференциала [Clarke, 1990].

Definition 3 (Обобщенный якобиан и Clarke-Гессиан)**.** *Для локально липшицевой функции* $f : \mathbb{R}^n \to \mathbb{R}^m$, *обобщенный якобиан по Кларку в точке* x *определяется как*

$$\partial_C f(x) = \mathrm{co}\{\lim_{i\to\infty} \nabla f(x_i) : x_i \to x, x_i \in \mathcal{D}_f\},$$

где co *— выпуклая оболочка, а* $\mathcal{D}_f$ *— множество точек, где* f *дифференцируема.*

Clarke-Гессиан для функции f *определяется как обобщенный якобиан градиента* ∇f *(если он существует):*

$$\partial_C^2 f(x) = \partial_C(\nabla f)(x).$$

Theorem 1 (Существование Clarke-Гессиана для ReLU-сетей)**.** *Пусть нейронная сеть использует активации* $\mathrm{ReLU}(t) = \max\{0, t\}$ *и имеет DAG вычислений* $G = (V, E)$. *Обозначим через* $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$ *функцию потерь, полученную как композицию сети* $F : \mathbb{R}^d \to \mathbb{R}^m$ *с внешней функцией* $\ell : \mathbb{R}^m \to \mathbb{R}$*:*

$$\mathcal{L}(x) = \ell\big(F(x)\big).$$

Предположим, что ℓ *локально липшицева. Тогда*

1. *Каждая функция узла* f_v *локально липшицева.*
2. *Во всех точках* x *Clarke-субдифференциал* $\partial_C \mathcal{L}(x)$ *непуст.*
3. *В любой точке* x*, где* $\mathcal{L}$ *дважды дифференцируема, Clarke-Гессиан* $\partial_C^2 \mathcal{L}(x)$ *вырождается в одиночное множество, совпадающее с обычным Гессианом* $\nabla^2 \mathcal{L}(x)$.
4. *В точках негладкости Clarke-Гессиан представляет собой непустое, выпуклое и компактное множество матриц.*

Proof. **Шаг 1 (локальная липшицевость каждого узла).** Рассмотрим топологический порядок вершин $v_1, \dots, v_{|V|}$. Для входного узла v_1 функция $f_{v_1}(x) = x$ очевидно 1-липшицева. Пусть узел v получает выходы $f_{u_1}, \dots, f_{u_k}$ предыдущих вершин и применяет линейное преобразование $W_v(\,\cdot\,) + b_v$, за которым следует ReLU:

$$f_v(x) = \mathrm{ReLU}\big(W_v\, [f_{u_1}(x), \dots, f_{u_k}(x)]^\top + b_v\big).$$

Линейное отображение имеет константу Липшица $\|W_v\|_2$, а ReLU — константу 1. Следовательно, f_v $\big(\prod_{i=1}^k L_{u_i}\big)\|W_v\|_2$-липшицева, где L_{u_i} — константа для f_{u_i}. Индукцией по порядку вершин получаем локальную липшицевость всех f_v.

Шаг 2 (непустота Clarke-субдифференциала). Композиция локально липшицевых отображений снова локально липшицева, значит $\mathcal{L}$ локально липшицева. По теореме Радемахера такая функция дифференцируема почти всюду, а по теореме 2.5.1 Кларка Clarke [1990] её Clarke-субдифференциал $\partial_C \mathcal{L}(x)$ непуст в каждой точке $x \in \mathbb{R}^d$.

Шаг 3 (совпадение Гессианов в гладких точках). Пусть x — точка, где $\mathcal{L}$ дважды дифференцируема. Тогда градиент $\nabla \mathcal{L}$ непрерывен в окрестности x и дифференцируем в x, так что по определению обобщённого Гессиана

$$\partial_C^2 \mathcal{L}(x) = \big\{\nabla^2 \mathcal{L}(x)\big\}.$$

Замечание: внутри одного линейного региона сети карта F аффинна, поэтому, если внешняя функция ℓ гладкая класса C^2, обычный Гессиан вычисляется стандартно; для квадратичной или кросс-энтропийной ℓ он, generally, ненулевой.

Шаг 4 (компактность и выпуклость в негладких точках). Рассмотрим точку x, где $\mathcal{L}$ не дважды дифференцируема. По определению

$$\partial_C^2 \mathcal{L}(x) = \operatorname{conv}\Big\{ \lim_{k\to\infty} \nabla^2 \mathcal{L}(x_k) \,:\, x_k \to x,\ x_k \text{ — гладкие точки} \Big\}.$$

Множество предельных матриц ограничено (поскольку каждая $\nabla^2 \mathcal{L}(x_k)$ зависит только от конечного набора линейных регионов и весов сети) и замкнуто; его выпуклая оболочка остаётся замкнутой и ограниченной, а значит компактной в конечномерном пространстве матриц (см. теорему 2.6.3 Кларка Clarke [1990]). Выпуклость следует из свойства «conv» по построению.

Таким образом, все четыре утверждения теоремы доказаны. □

Definition 4 (Clarke-Гессиан с минимальной нормой)**.** *В негладком случае (Случай B), вместо единственного блока $H^f_{v,v}$ и соответствующих H_{θ_v,θ_w} получаем множество $\partial_C^2 f_v$. Конкретный элемент этого множества выбирается из условия минимальной нормы Фробениуса:*

$$H^f_{v,w} = \arg\min_{M\in\partial_C^2 f_v} \|M\|_F, \quad H_{\theta_v,\theta_w} = \arg\min_{M\in\partial^2_{\theta_v,\theta_w}\mathcal{L}} \|M\|_F.$$

Такой выбор обеспечивает единственность элемента субдифференциала и соответствует принципу минимальной нормы, часто используемому в недифференцируемой оптимизации [Bolte and Pauwels, 2020].

3.7 Полный входной Гессиан

Definition 5 (Входной Гессиан)**.** *Полный входной Гессиан представляет собой блочную матрицу $\{H^f_{v,w}\}_{v,w\in V}$, где каждый блок $H^f_{v,w} \in \mathbb{R}^{d_v\times d_w}$ определяется рекурсивно:*

$$\boxed{\begin{aligned} H^f_{v,w} = &\sum_{u\in \mathrm{Ch}(v)\cap \mathrm{Ch}(w)} D_{u\leftarrow v}^\top\, H^f_{u,u}\, D_{u\leftarrow w} \quad \text{(Гаусс-Ньютон)} \\ &+ \sum_{u\in \mathrm{Ch}(v)\cap \mathrm{Ch}(w)} \sum_{i=1}^{d_u} [T_{u;\,v,w}]_{i,\bullet,\bullet}\, \delta_{u,i} \quad \text{(смешанные входы)} \\ &+ \delta_{v=w} \sum_{u\in \mathrm{Ch}(v)} \sum_{i=1}^{d_u} [T_{u;v}]_{i,\bullet,\bullet}\, \delta_{u,i} \quad \text{(чистые по одному входу)} \end{aligned}} \tag{1}$$

с базовыми условиями:

$$H^f_{out,out} = \nabla^2 \mathcal{L}(f_{out}), \quad H^f_{out,v} = H^f_{v,out} = 0 \quad (\forall v \neq out),$$

$$H^f_{v,w} = 0 \quad \text{если нет узла } u, \text{ являющегося потомком и } v, \text{ и } w.$$

Свойство симметрии: $H^f_{v,w} = (H^f_{w,v})^\top$ для всех $v, w \in V$.

Первый член в уравнении (1) соответствует классической части Гаусса-Ньютона, которая учитывает только первые производные функций узлов. Второй член учитывает смешанные вторые производные между различными входами, а третий член (активный только при $v = w$) добавляет вклад от чистых вторых производных по одному входу.

3.8 Полный параметрический Гессиан

Definition 6 (Параметрический Гессиан)**.** *Полный Гессиан по параметрам* $\nabla^2_\theta \mathcal{L}$ *разбивается на блоки* $\{H_{\theta_v,\theta_w}\}$, $H_{\theta_v,\theta_w} \in \mathbb{R}^{p_v \times p_w}$, *каждый из которых определяется как:*

$$\boxed{\begin{aligned} H_{\theta_v,\theta_w} = {} & D_v^\top \, H^f_{v,w} \, D_w \quad \text{(Гаусс-Ньютон)} \\ & + \delta_{v=w} \sum_{i=1}^{d_v} [T_v^\theta]_{i,\bullet,\bullet} \, \delta_{v,i} \quad \text{(чистые по параметрам)} \\ & + \sum_{u \in \mathrm{Pa}(v) \cap \mathrm{Ch}(w)} \sum_{i=1}^{d_v} \sum_{j=1}^{d_u} \sum_{k=1}^{p_v} [T_{v;\,u,\theta}]_{i,j,k} \, (D_{w \leftarrow u})_{j,i} \, \delta_{v,i} \quad \text{(смешанные вход-параметр } v \to w) \\ & + \sum_{u \in \mathrm{Pa}(w) \cap \mathrm{Ch}(v)} \sum_{i=1}^{d_w} \sum_{j=1}^{d_u} \sum_{\ell=1}^{p_w} [T_{w;\,u,\theta}]_{i,j,\ell} \, (D_{v \leftarrow u})_{j,i} \, \delta_{w,i} \quad \text{(смешанные вход-параметр } w \to v) \end{aligned}} \tag{2}$$

Remark 3 (О суммировании в формуле (2))**.** *Для корректного понимания смешанных сумм в третьей и четвертой строках формулы* (2) *важно уточнить индексацию:*

- *В выражении* $[T_{v;\,u,\theta}]_{i,j,k}$ *индекс* i *относится к компонентам выхода* f_v *узла* v, *индекс* j *- к компонентам входа от узла* u, *а индекс* k *- к компонентам параметра* θ_v.
- *Матрица* $(D_{w \leftarrow u})_{j,i}$ *имеет индексы, где* j *соответствует входным компонентам узла* w *от узла* u, *а* i *- выходным компонентам узла* w.
- *Суммирование производится по всем компонентам* i *выходов узла* v, *всем компонентам* j *входов от узла* u *и всем компонентам* k *параметра* θ_v.
- *Аналогично интерпретируются индексы в четвертой строке, с заменой ролей* v *и* w.

Такая структура суммирования обеспечивает согласованное агрегирование всех смешанных вторых производных по входам и параметрам, соответствующих вкладам от всех путей в графе нейронной сети.

Базовое условие для офф-диагональных блоков:

$$H_{\theta_v,\theta_w} = 0 \quad \text{если не существует путей } v \to u \text{ и } w \to u.$$

Свойство симметрии: $H_{\theta_v,\theta_w} = (H_{\theta_w,\theta_v})^\top$.

Первый член в уравнении (2) представляет Гаусс-Ньютон часть Гессиана. Второй член (активен только при $v = w$) учитывает чистые вторые производные по параметрам. Третий и четвёртый члены отвечают за смешанные производные между входами и параметрами для различных конфигураций узлов.

3.9 Разделение параметров между узлами

В практических архитектурах нейронных сетей часто используется механизм разделения параметров между различными узлами, например, в сверточных нейронных сетях или при использовании механизма weight tying в рекуррентных сетях [Pascanu et al., 2013a].

Proposition 2 (Гессиан разделяемых параметров)**.** *Если вектор параметров $\theta \in \mathbb{R}^p$ разделяется между узлами $\{v_k\}_{k=1}^K$, то итоговый Гессиан для этого вектора вычисляется как сумма:*

$$H_{\theta,\theta} = \sum_{a=1}^{K}\sum_{b=1}^{K} H_{\theta_{v_a},\theta_{v_b}}.$$

Это правило учитывает все возможные взаимодействия между параметрами, как внутри одного узла, так и между различными узлами, использующими один и тот же вектор параметров.

4 Алгоритмы вычисления

4.1 Общий алгоритм вычисления полного Гессиана

5 Теоретические результаты

5.1 Функционально-аналитические свойства Гессиана

Theorem 2 (Функционально-аналитические свойства Гессиана)**.** *При выполнении Предположения 1 о регулярности функций узлов, полный Гессиан $\nabla^2_\theta \mathcal{L}$ обладает следующими свойствами:*

Algorithm 1 Вычисление полного Гессиана для нейронной сети

Require: Нейронная сеть с DAG $G = (V, E)$, функции узлов $\{g_v\}$, параметры $\{\theta_v\}$, функция потерь $\mathcal{L}$
Ensure: Полный Гессиан $\nabla^2_\theta \mathcal{L}$
Вычислить прямой проход и получить f_v для всех $v \in V$
Вычислить $\delta_{out} = \nabla_{f_{out}} \mathcal{L}$ и $H^f_{out,out} = \nabla^2 \mathcal{L}(f_{out})$
Инициализировать $H^f_{v,w} = 0$ для всех пар $v, w \in V$, $v \neq out$, $w \neq out$
for $v \in V$ в обратном топологическом порядке **do**
    Вычислить δ_v по цепному правилу
    for $w \in V$ такие, что $\mathrm{Ch}(v) \cap \mathrm{Ch}(w) \neq \emptyset$ **do**
        Вычислить $H^f_{v,w}$ по формуле (1)
    end for
end for
for $v \in V$ **do**
    for $w \in V$ такие, что существуют пути $v \to u$ и $w \to u$ **do**
        Вычислить H_{θ_v,θ_w} по формуле (2)
    end for
end for
Учесть разделение параметров между узлами
Собрать блоки в полный Гессиан $\nabla^2_\theta \mathcal{L}$
return $\nabla^2_\theta \mathcal{L}$

1. *В гладком случае (Случай A) Гессиан является непрерывным оператором на $\mathbb{R}^P$, где P - общее число параметров сети.*

2. *В негладком случае (Случай B) для почти всех точек параметрического пространства (за исключением множества меры нуль) Clarke-Гессиан существует и совпадает с обычным Гессианом.*

3. *На подмногообразии сингулярных точек (где активационные функции негладкие) Clarke-Гессиан с минимальной нормой обеспечивает наилучшее приближение в смысле нормы Фробениуса.*

4. *При использовании предложенных формул* (1) *и* (2) *обеспечивается согласованность размерностей всех тензорных операций.*

5.2 Интеграция специализированных архитектурных компонентов

Theorem 3 (Интеграция специализированных слоёв)**.** *Следующие архитектитектурные компоненты могут быть представлены в виде узлов DAG и включены в предложенный формализм:*

1. ***Batch Normalization****: представляется как узел с двумя типами параметров (маситабирующие и сдвиговые) и дополнительными внутренними переменными (статистики батча).*

2. ***Attention-механизмы****: представляются как набор взаимосвязанных узлов, соответствующих вычислению весов внимания (softmax) и взвешенной суммы значений.*

3. ***Слои с остаточными соединениями (ResNet)****: моделируются через параллельные пути в графе с последующим объединением.*

4. ***Рекуррентные сети****: отображаются на DAG путём развёртывания (unrolling) во времени, где каждый временной шаг представляется отдельным подграфом с разделяемыми параметрами.*

Схема доказательства. Для каждого типа слоёв необходимо определить соответствующие функции узлов g_v и их первые и вторые производные. Например, для Batch Normalization:

$$g_v(x, \gamma, \beta) = \gamma \frac{x - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} + \beta$$

где μ_B, σ_B^2 — средние и дисперсии по батчу, γ, β — параметры масштаба и сдвига.

Якобианы D_v и тензоры вторых производных T_v вычисляются по стандартным правилам дифференцирования для каждого типа узлов, после чего применяются общие формулы (1) и (2). $\square$

5.3 Стохастические узлы и вариационные подходы

Definition 7 (Стохастический узел). *Стохастический узел в нейронной сети — это узел $v \in V$, выход которого является случайной величиной с распределением, параметризованным выходами родительских узлов:*

$$f_v \sim p(f_v | f_{\mathrm{Pa}(v)}, \theta_v)$$

Theorem 4 (Гессиан со стохастическими узлами). *Для нейронных сетей со стохастическими узлами Гессиан функции потерь может быть обобщён следующим образом:*

1. *При использовании подхода максимального правдоподобия формулы* (1) *и* (2) *применяются к ожидаемой функции потерь* $\mathbb{E}_{f_v \sim p}[\mathcal{L}]$.
2. *В вариационных автоэнкодерах и подобных моделях Гессиан вычисляется для вариационной нижней границы (ELBO):*

$$\mathcal{L}_{ELBO} = \mathbb{E}_{q(z|x)}[\log p(x|z)] - D_{KL}(q(z|x)||p(z))$$

3. *Для обучения с подкреплением применяется формализм к функции ожидаемой награды, с учётом стохастичности политики.*

Proposition 3 (Переключение к детерминированным узлам). *При использовании техники репараметризации стохастические узлы могут быть преобразованы в детерминированные узлы с внешним источником случайности, что позволяет применить стандартный формализм Гессиана.*

6 Анализ вычислительной сложности

Theorem 5 (Вычислительная сложность). *Пусть* $|V| = n$ *— число узлов в DAG,* $P = \sum_{v \in V} p_v$ *— общее число параметров,* $d = \max_{v \in V} d_v$ *— максимальная размерность выхода узла,* $k = \max_{v \in V} |\mathrm{Pa}(v)|$ *— максимальное число входов узла. Тогда:*

1. *Временная сложность вычисления полного Гессиана составляет* $O(n^2 d^3 + n^2 d^2 P + P^2)$.
2. *Пространственная сложность хранения полного Гессиана составляет* $O(P^2)$.
3. *Для прореженного DAG с максимальной степенью узла* $s \ll n$ *временная сложность снижается до* $O(nsd^3 + nsd^2 P + P^2)$.

Схема доказательства. Для входного Гессиана $H^f_{v,w}$ формула (1) требует:

- $O(|\mathrm{Ch}(v) \cap \mathrm{Ch}(w)| \cdot d^3)$ операций для вычисления Гаусс-Ньютон части.
- $O(|\mathrm{Ch}(v) \cap \mathrm{Ch}(w)| \cdot d^3)$ операций для вычисления тензорных членов.

Существует $O(n^2)$ пар узлов (v, w), но для прореженных графов число пар с непустым пересечением $\mathrm{Ch}(v) \cap \mathrm{Ch}(w)$ может быть $O(ns)$, где s — максимальная степень узла.

Для параметрического Гессиана H_{θ_v,θ_w} формула (2) требует:

- $O(d_v \cdot d_w \cdot p_v \cdot p_w)$ операций для вычисления Гаусс-Ньютон части.
- Дополнительные $O(d \cdot p_v \cdot p_w)$ операции для тензорных членов.

Суммирование по всем парам (v, w) даёт общую сложность $O(P^2)$ для сборки полного параметрического Гессиана. □

Theorem 6 (Методы снижения вычислительных затрат)**.** *Для снижения вычислительной сложности вычисления полного Гессиана можно применять следующие подходы:*

1. ***Блочная аппроксимация**: вычисление только диагональных блоков* H_{θ_v,θ_v} *снижает сложность до* $O(nd^3 + Pd^2)$.
2. ***Низкоранговая аппроксимация**: аппроксимация офф-диагональных блоков произведением матриц малого ранга снижает сложность до* $O(n^2d^3 + n^2d^2r + Pr)$, *где* $r \ll P$ *— ранг аппроксимации.*
3. ***Гаусс-Ньютон аппроксимация**: использование только первого члена в формулах (1) и (2) снижает сложность и гарантирует положительную полуопределённость.*
4. ***Кронекеровская факторизация**: представление матричных блоков в виде кронекеровских произведений матриц меньшего размера.*

7 Анализ сходимости методов оптимизации

Theorem 7 (Локальная сходимость методов Ньютона)**.** *Пусть* $\mathcal{L}(\theta) \in C^2$ *— функция потерь, и* θ^* *— её локальный минимум, такой что* $\nabla_\theta^2 \mathcal{L}(\theta^*) \succ 0$. *Тогда метод Ньютона со степенным шагом:*

$$\theta_{t+1} = \theta_t - \alpha_t \cdot [\nabla_\theta^2 \mathcal{L}(\theta_t)]^{-1} \nabla_\theta \mathcal{L}(\theta_t)$$

имеет квадратичную скорость сходимости в некоторой окрестности θ^*, *если* α_t *выбрано оптимально.*

Proposition 4 (Критерии остановки)**.** *Учитывая структуру Гессиана в нейронных сетях, можно разработать следующие критерии остановки для оптимизационных алгоритмов:*

1. *Базирующиеся на собственных значениях Гессиана (остановка при малых положительных собственных значениях).*
2. *Использующие относительную норму градиента:* $\|\nabla_\theta \mathcal{L}(\theta_t)\| / \|\nabla^2_\theta \mathcal{L}(\theta_t)\| < \epsilon$.
3. *Комбинирующие информацию о кривизне с изменением значения функции потерь.*

Theorem 8 (Гарантии сходимости для регуляризованных методов)**.** *Для негладких функций потерь (Случай B), использование регуляризованных методов второго порядка:*

$$\theta_{t+1} = \theta_t - (H_t + \lambda I)^{-1} \nabla_\theta \mathcal{L}(\theta_t),$$

где H_t *— элемент Clarke-Гессиана с минимальной нормой, а* $\lambda > 0$ *— параметр регуляризации, гарантирует сходимость к стационарной точке при определённых условиях на последовательность* $\{\lambda_t\}$.

8 Результаты и обсуждение

8.1 Практические замечания

При практической реализации вычисления полного Гессиана необходимо учитывать следующие аспекты:

- В **гладком случае** рекомендуется проверять положительную полуопределённость Гаусс-Ньютон части $D_v^\top H^f_{v,v} D_v$ перед добавлением остальных слагаемых. Это позволяет обеспечить стабильность методов оптимизации, основанных на Гессиане.
- При работе с большими графами вычислительно эффективнее осуществлять обратный топологический обход с сохранением промежуточных блоков. Такой подход позволяет избежать повторных вычислений и значительно ускоряет процесс построения полного Гессиана.

8.2 Сравнение с существующими подходами

Предложенный формализм существенно расширяет традиционные подходы к анализу кривизны функций потерь нейронных сетей:

1. **Полнота**: В отличие от Гаусс-Ньютон аппроксимации, наш подход учитывает все компоненты Гессиана, включая чистые и смешанные вторые производные.

2. **Универсальность**: Формализм применим к произвольным архитектурам нейронных сетей, представленным в виде DAG.

3. **Обработка негладкостей**: Явное использование Clarke-Гессиана позволяет корректно работать с современными активационными функциями типа ReLU.

4. **Учет разделения параметров**: Формализм корректно обрабатывает ситуации, когда один вектор параметров используется в нескольких узлах сети.

9 Заключение

В данной работе представлен исчерпывающий математический формализм для вычисления полного Гессиана второго порядка в нейронных сетях произвольной архитектуры. Основные достижения работы:

- Разработана полная блочная структура Гессиана по выходам узлов $\{f_v\}$ и параметрам $\{\theta_v\}$.
- Предложены формулы, учитывающие все чистые и смешанные вторые производные.
- Определены явные базовые случаи для офф-диагональных блоков.
- Сформулированы отдельные правила для гладкого и негладкого случаев.
- Представлено математическое обоснование для тривиального предельного случая с одним узлом.
- Даны практические рекомендации по проверке положительной полуопределенности Гаусс-Ньютон части.

Предложенный формализм создает теоретическую основу для разработки более эффективных методов оптимизации нейронных сетей, глубокого анализа кривизны функций потерь и понимания геометрической структуры пространства параметров. Дальнейшие исследования могут быть направлены на разработку вычислительно эффективных аппроксимаций полного Гессиана и использование полученной информации о кривизне в алгоритмах оптимизации нейронных сетей произвольной структуры.

References

Amari, S.-I. (1998). Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276.

Bolte, J. and Pauwels, E. (2020). Conservative set valued fields, automatic differentiation, stochastic gradient methods and deep learning. *Mathematical Programming*, pp. 1–33.

Clarke, F. H. (1990). *Optimization and nonsmooth analysis*, volume 5. Siam.

Ghorbani, B., Krishnan, S., and Xiao, Y. (2019). An investigation into neural net optimization via hessian eigenvalue density. In *International Conference on Machine Learning*, pp. 2232–2241.

Heskes, T. (2000). On natural learning and pruning in multilayered perceptrons. *Neural Computation*, 12(4):881–901.

Martens, J. (2010). Deep learning via hessian-free optimization. In *ICML*, volume 27, pp. 735–742.

Martens, J. (2014). New insights and perspectives on the natural gradient method. *arXiv preprint arXiv:1412.1193*.

Martens, J. and Sutskever, I. (2012). Training deep and recurrent networks with hessian-free optimization. In *Neural networks: Tricks of the trade*, pp. 479–535. Springer.

Nocedal, J. and Wright, S. (2006). *Numerical optimization*. Springer Science & Business Media.

Pascanu, R., Mikolov, T., and Bengio, Y. (2013a). On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pp. 1310–1318.

Pascanu, R., Montufar, G., and Bengio, Y. (2013b). On the number of response regions of deep feed forward networks with piece-wise linear activations. *arXiv preprint arXiv:1312.6098*.

Sagun, L., Evci, U., Guney, V. U., Dauphin, Y., and Bottou, L. (2017). Empirical analysis of the hessian of over-parametrized neural networks. *arXiv preprint arXiv:1706.04454*.

Schraudolph, N. N. (2002). Fast curvature matrix-vector products for second-order gradient descent. *Neural computation*, 14(7):1723–1738.

A Реализация с использованием autodiff-фреймворков

Алгоритм 1: Вычисление блока входного Гессиана

function $\textsc{ComputeInputHessian}(v,\ w,\ \{f_u\},\ \mathcal{L},\ \{\delta_u\},\ \{H^f_{u,u'}\})$
  $H^f_{v,w} \leftarrow 0$ ▷ Инициализация блока входного Гессиана
  for $u \in \mathrm{Ch}(v) \cap \mathrm{Ch}(w)$ **do**
    $D_{u\leftarrow v} \leftarrow$ autodiff.jacobian(f_u, f_v)
    $D_{u\leftarrow w} \leftarrow$ autodiff.jacobian(f_u, f_w)
    $H^f_{v,w} \leftarrow H^f_{v,w} + D^\top_{u\leftarrow v} H^f_{u,u} D_{u\leftarrow w}$
    for $i \in 1..d_u$ **do**
      $T_{u;v,w} \leftarrow$ autodiff.hessian_mixed$(f_{u,i}, f_v, f_w)$
      $H^f_{v,w} \leftarrow H^f_{v,w} + T_{u;v,w} \cdot \delta_{u,i}$
    end for
    if $v = w$ **then**
      for $i \in 1..d_u$ **do**
        $T_{u;v} \leftarrow$ autodiff.hessian$(f_{u,i}, f_v)$
        $H^f_{v,v} \leftarrow H^f_{v,v} + T_{u;v} \cdot \delta_{u,i}$
      end for
    end if
  end for
  return $H^f_{v,w}$
end function

Алгоритм 2: Вычисление блока параметрического Гессиана

function $\textsc{ComputeParameterHessian}(v,\ w,\ \{f_u\},\ \{\theta_u\},\ \{H^f_{u,u'}\},\ \{\delta_u\})$
  $D_v \leftarrow$ autodiff.jacobian(f_v, θ_v)
  $D_w \leftarrow$ autodiff.jacobian(f_w, θ_w)
  $H_{\theta_v,\theta_w} \leftarrow D^\top_v H^f_{v,w} D_w$
  if $v = w$ **then**
    for $i \in 1..d_v$ **do**
      $T^\theta_v \leftarrow$ autodiff.hessian$(f_{v,i}, \theta_v)$
      $H_{\theta_v,\theta_v} \leftarrow H_{\theta_v,\theta_v} + T^\theta_v \cdot \delta_{v,i}$
    end for
  end if
  for $u \in \mathrm{Pa}(v) \cap \mathrm{Ch}(w)$ **do**
    for $i \in 1..d_v$ **do**
      for $j \in 1..d_u$ **do**
        for $k \in 1..p_v$ **do**
          $T_{v;u,\theta} \leftarrow$ autodiff.mixed_derivative$(f_{v,i}, f_{u,j}, \theta_{v,k})$
          $D_{w\leftarrow u} \leftarrow$ autodiff.jacobian(f_w, f_u)
          $H_{\theta_v,\theta_w} \leftarrow H_{\theta_v,\theta_w} + T_{v;u,\theta} \cdot D_{w\leftarrow u} \cdot \delta_{v,i}$
        end for
      end for
    end for
  end for
  return H_{θ_v,θ_w}
end function

Алгоритм 3: Полное вычисление Гессиана

function FullHessianComputation(G, $\{f_v\}$, $\{\theta_v\}$, $\mathcal{L}$)
    $\delta_{out} \leftarrow$ autodiff.gradient($\mathcal{L}, f_{out}$)
    $H^f_{out,out} \leftarrow$ autodiff.hessian($\mathcal{L}, f_{out}$)
    topo_order $\leftarrow$ TopologicalSort(G).reverse()
    Initialize $\{\delta_v\}, \{H^f_{v,w}\}$ as zero matrices
    for $v \in$ topo_order **do**
        BackpropagateGradients(v)
        for $w \in V$ such that $\mathrm{Ch}(v) \cap \mathrm{Ch}(w) \neq \emptyset$ **do**
            $H^f_{v,w} \leftarrow$ ComputeInputHessian($v, w, \{f_u\}, \mathcal{L}, \{\delta_u\}, \{H^f_{u,u'}\}$)
        end for
    end for
    Initialize full Hessian matrix H of size $P \times P$
    for $v, w \in V$ **do**
        if $\exists u : v \to u$ и $w \to u$ **then**
            $H_{\theta_v,\theta_w} \leftarrow$ ComputeParameterHessian($v, w, \{f_u\}, \{\theta_u\}, \{H^f_{u,u'}\}, \{\delta_u\}$)
            Update corresponding blocks in H
        end if
    end for
    return H
end function

Remark 4 (Реализация в фреймворках машинного обучения)**.** *Представленный псевдокод может быть реализован с использованием механизмов автоматического дифференцирования:*

- *В PyTorch — с использованием функций* `torch.autograd.grad` *и* `torch.autograd.functional.jacobian, hessian`
- *В TensorFlow — с использованием* `tf.GradientTape` *с вложенными вызовами для вычисления высших производных*
- *В JAX — с использованием функций* `jax.grad`*,* `jax.jacobian` *и* `jax.hessian`

Для работы с разреженными структурами и оптимизации вычислений рекомендуется использовать специализированные библиотеки, такие как SciPy (для разреженных матриц) или соответствующие расширения фреймворков.