

Применение многослойного перцептрона к решению многомерных задач квантовой механики

22 октября 2022 г.

1 Вычисление оператора Лапласа для многослойного перцептрона

1.1 Математическое описание многослойного перцептрона

Пусть на вход перцептрону подаётся вектор $\mathbf{x} \in \mathbb{R}^n$, он последовательно линейно преобразуется каждым слоем и в результате на выходе перцептрона имеется вектор $\mathbf{y} \in \mathbb{R}^m$. Пусть l — число внутренних слоёв нейронной сети, а на k -ом слое имеется n_k нейронов ($n_0 = n$, $n_{l+1} = m$). Через $\mathbf{h}^{(k)} \in \mathbb{R}^{n_k}$ будет обозначаться вектор на выходе с k -ого слоя. Используя введённые обозначения можно связать \mathbf{x} и \mathbf{y} следующим образом:

$$\begin{aligned} h_i^{(0)} &= x_i, \quad i = \overline{1, n_0}; \\ h_i^{(1)} &= f^{(1)} \left(\sum_{j=1}^{n_0} W_{i,j}^{(1)} h_j^{(0)} + b_i^{(1)} \right), \quad i = \overline{1, n_1}; \\ &\dots \\ h_i^{(k)} &= f^{(k)} \left(\sum_{j=1}^{n_{k-1}} W_{i,j}^{(k)} h_j^{(k-1)} + b_i^{(k)} \right), \quad i = \overline{1, n_k}; \\ &\dots \\ h_i^{(l)} &= f^{(l)} \left(\sum_{j=1}^{n_{l-1}} W_{i,j}^{(l)} h_j^{(l-1)} + b_i^{(l)} \right), \quad i = \overline{1, n_l}; \\ y_i = h_i^{(l+1)} &= f^{(l+1)} \left(\sum_{j=1}^{n_l} W_{i,j}^{(l+1)} h_j^{(l)} + b_i^{(l+1)} \right), \quad i = \overline{1, n_{l+1}}. \end{aligned} \tag{1.1}$$

В предыдущей записи функция $f^{(k)}$ называется функцией активации k -го слоя, матрица $\hat{W}^{(k)}$ — матрицей весов размера $n_k \times n_{k-1}$ (первый индекс — номер строки, второй — номер столбца), а вектор $b^{(k)}$ — вектором сдвижки.

1.2 Первая производная многослойного перцептрона

Продифференцируем перцептрон по x_t , $t = \overline{1, n_0}$. Из (1.1) и правил взятия сложной производной следует такая схема расчёта

$$\begin{aligned} \frac{\partial y_i}{\partial x_t} &= \frac{\partial h_i^{(l+1)}}{\partial x_t}, \quad i = \overline{1, n_{l+1}}; \\ \frac{\partial h_i^{(k)}}{\partial x_t} &= f^{(k)'} \left(\sum_{j=1}^{n_{k-1}} W_{i,j}^{(k)} h_j^{(k-1)} + b_i^{(k)} \right) \cdot \sum_{j=1}^{n_{k-1}} W_{i,j}^{(k)} \frac{\partial h_j^{(k-1)}}{\partial x_t}, \quad i = \overline{1, n_k}, \quad k = \overline{1, l+1}; \\ \frac{\partial h_i^{(0)}}{\partial x_t} &= \delta_{i,t}, \quad i = \overline{1, n_0}. \end{aligned} \quad (1.2)$$

Можно заметить, что для вычисления производной разумно идти от нижних слоёв к верхним, так как вычисление каждой последующей $\partial h_i^{(k)} / \partial x_t$ зависит от значения таких производных на предыдущих слоях.

1.3 Лапласиан многослойного перцептрона

Основываясь на предыдущих результатах и полагая, что вектор \mathbf{x} дан в n -мерной декартовой системе координат, нетрудно получить схему вычисления лапласиана перцептрона, которая имеет следующий вид:

$$\nabla^2 y_i = \sum_{t=1}^{n_0} \frac{\partial^2 y_i}{\partial x_t^2} = \sum_{t=1}^{n_0} \frac{\partial^2 h_i^{(l+1)}}{\partial x_t^2}, \quad i = \overline{1, n_{l+1}}; \quad (1.3)$$

$$\begin{aligned} \frac{\partial^2 h_i^{(k)}}{\partial x_t^2} &= f^{(k)''} \left(\sum_{j=1}^{n_{k-1}} W_{i,j}^{(k)} h_j^{(k-1)} + b_i^{(k)} \right) \cdot \left\{ \sum_{j=1}^{n_{k-1}} W_{i,j}^{(k)} \frac{\partial h_j^{(k-1)}}{\partial x_t} \right\}^2 \\ &+ f^{(k)'} \left(\sum_{j=1}^{n_{k-1}} W_{i,j}^{(k)} h_j^{(k-1)} + b_i^{(k)} \right) \cdot \sum_{j=1}^{n_{k-1}} W_{i,j}^{(k)} \frac{\partial^2 h_j^{(k-1)}}{\partial x_t^2}, \quad i = \overline{1, n_k}, \quad k = \overline{1, l+1}; \end{aligned} \quad (1.4)$$

$$\frac{\partial^2 h_i^{(0)}}{\partial x_t^2} = 0, \quad i = \overline{1, n_0}. \quad (1.5)$$

При вычислении лапласиана, как и при вычислении градиента, следует начинать с нижних слоёв.

1.4 Программная реализация и верификация

Алгоритм вычисления градиента и лапласиана для многослойного перцептрона был реализован на языке Python с использованием библиотеки PyTorch. Вариант реализации и его верификация, описанная ниже, доступны по ссылке.

Для оценки верности работы алгоритмов был рассмотрен пример перцептрона с одним внутренним слоем, содержащим n нейронов. На вход такой нейронной сети подавался вектор $\mathbf{x} \in \mathbb{R}^n$, что позволило выбрать веса следующими

$$\hat{W}^{(1)} = \hat{I}, \quad \hat{W}_i^{(2)} = 1 \quad \forall i. \quad (1.6)$$

Тогда нетрудно получить выражение результата работы перцептрона по входному вектору

$$y(\mathbf{x}) = \tanh \left(\sum_{k=1}^n \tanh(x_k) \right). \quad (1.7)$$

Отсюда можно получить выражение для градиента перцептрона

$$\nabla y(\mathbf{x}) = \sum_{j=1}^n \cosh^{-2} \left(\sum_{k=1}^n \tanh(x_k) \right) \cosh^{-2}(x_j) \mathbf{e}_j, \quad (1.8)$$

где вектора $\{\mathbf{e}_j\}_{j=1}^n$ образуют стандартный базис в пространстве \mathbb{R}^n . Выражение для лапласиана функции $y(\mathbf{x})$ удаётся записать, используя предыдущие производные

$$\nabla^2 y(\mathbf{x}) = \sum_{i=1}^n \frac{\partial y}{\partial x_i}(\mathbf{x}) \left(\tanh(x_i) + \frac{y(\mathbf{x})}{\cosh^2(x_i)} \right). \quad (1.9)$$

Результаты тестирования доступны по ссылке. Из них можно заключить, что значение нейронной сети, её градиента и лапласиана вычисляются верно.

2 Вычисление спектра N-мерного оператора Лапласа с помощью многослойного перцептрона

2.1 Квантово-механическая постановка задачи

Рассмотрим стационарное уравнение Шрёдингера (УШ) для свободного электрона в N-мерной потенциальной яме с бесконечными стенками

$$\hat{H} \psi(\mathbf{x}) = -\frac{\hbar^2}{2m} \nabla^2 \psi(\mathbf{x}) + V(\mathbf{x}) \psi(\mathbf{x}) = E \psi(\mathbf{x}), \quad (2.1)$$

где $\mathbf{x} \in \mathbb{R}^N$, а потенциал задаётся формулой

$$V(\mathbf{x}) = \begin{cases} 0, & x_i \in (0, L), i = \overline{1, N}; \\ +\infty, & \text{иначе.} \end{cases} \quad (2.2)$$

Очевидно, что такая задача сводится к задаче отыскания спектра N-мерного оператора Лапласа с нулевыми граничными условиями на сторонах N-мерного куба

$$-\nabla^2 \psi(\mathbf{x}) = \frac{2mE}{\hbar^2} \psi(\mathbf{x}), \quad \psi(\mathbf{x}) \Big|_{x_i=0, L} = 0, i = \overline{1, N}. \quad (2.3)$$

Для краткости далее собственные значения оператора лапласа будут обозначаться через λ , которые связаны с энергией состояния электрона соотношением

$$\lambda = \frac{2mE}{\hbar^2}.$$

В итоге получаем задачу на собственные значения и собственные функции

$$\nabla^2 \psi(\mathbf{x}) = -\lambda \psi(\mathbf{x}) \quad (2.4)$$

с граничными условиями

$$\psi(\mathbf{x}) \Big|_{x_i=0, L} = 0, i = \overline{1, N}. \quad (2.5)$$

2.2 Неоднозначность подхода к решению задачи

Предлагается найти состояние с наименьшим собственным значением уравнения (2.4) с граничными условиями (2.5), используя для приближения нужной собственной функции оператора Лапласа перцептрон. Такая постановка задачи приводит к неоднозначному выбору функционала, минимум которого обеспечит наиболее подходящие параметры перцептрона, (то есть к неоднозначности выбора функции потерь) и к неоднозначности способа учёта граничных условий.

2.2.1 Способы учёта граничных условий

Учесть граничные условия (2.5) можно несколькими способами. Рассмотрим некоторые из них.

1. Регуляризация волновой функции.

Прежде всего можно регуляризовать волновую функцию ψ , рассмотрев её в виде произведения некоторой функции, удовлетворяющей граничным условиям, на перцептрон

$$\psi(\mathbf{x}) = A(\mathbf{x})P(\mathbf{x}), \quad A(\mathbf{x}) \Big|_{x_i=0, L} = 0, \quad i = \overline{1, N}, \quad (2.6)$$

где P — перцептрон.

2. Регуляризация функции потерь.

Можно также регуляризоваться и функцию потерь, добавив в неё член, минимальность которого обеспечивает минимальность значений волновой функции на границах рассматриваемой области. То есть предлагается перейти от функции потерь $I[\psi]$ к функции

$$I[\psi] + \alpha \sum_{q=1}^Q \left| \psi(\mathbf{x}^{(q)}) \right|, \quad \left\{ \mathbf{x}^{(q)} \right\}_{q=1}^Q \in \partial\Omega \quad (2.7)$$

или к функции

$$I[\psi] + \alpha \sum_{q=1}^Q \left| \psi(\mathbf{x}^{(q)}) \right|^2, \quad \left\{ \mathbf{x}^{(q)} \right\}_{q=1}^Q \in \partial\Omega, \quad (2.8)$$

где Ω — рассматриваемая область, $\partial\Omega$ — её граница, а коэффициент α определяет важность учёта граничных условий по сравнению с тем, что учитывается в $I[\psi]$.

2.2.2 Варианты задания функции потерь

Рассмотрим некоторые варианты задания функции потерь.

1. Функционал энергии.

В качестве функции потерь можно рассматривать функционал энергии $\langle \psi | \hat{H} | \psi \rangle / \langle \psi | \psi \rangle$ (в нашем случае $\hat{H} = -\nabla^2$). В случае многомерной задачи приближенно вычислять функционал энергии удобно с помощью метода Монте-Карло. Для этого поработаем с функцией потерь

$$\begin{aligned} I[\psi] &= \frac{\langle \psi | \hat{H} | \psi \rangle}{\langle \psi | \psi \rangle} = \frac{\int \psi^* \hat{H} \psi \, d\mathbf{x}}{\int |\psi|^2 \, d\mathbf{x}} = \int \frac{|\psi|^2}{\int |\psi|^2 \, d\mathbf{x}} \frac{\hat{H} \psi}{\psi} \, d\mathbf{x} \\ &= \int P[\psi, \mathbf{x}] E_{\text{loc}}[\psi, \mathbf{x}] \, d\mathbf{x} \approx \frac{1}{M} \sum_{m=1}^M E_{\text{loc}}[\psi, \mathbf{x}^{(m)}], \end{aligned} \quad (2.9)$$

где $P[\psi, \mathbf{x}] = |\psi|^2 / \int |\psi|^2 d\mathbf{x}$ — плотность вероятности, $E_{\text{loc}}[\psi, \mathbf{x}] = \hat{H}\psi/\psi$ — локальная энергия, а приближенное равенство означает предположение о равномерности плотности вероятности $P[\psi, \mathbf{x}] = 1/\int 1 d\mathbf{x}$. Таким образом, в качестве функции потерь можно использовать среднее значение локальной энергии.

2. Невязка.

Можно ещё в качестве функции потерь использовать невязку, которая задается как

$$I[\psi] = \sum_{m=1}^M \left| (\hat{H} - E)\psi(\mathbf{x}^{(m)}) \right| \quad (2.10)$$

или как

$$I[\psi] = \sum_{m=1}^M \left| (\hat{H} - E)\psi(\mathbf{x}^{(m)}) \right|^2, \quad (2.11)$$

где в нашем случае $\hat{H} = -\nabla^2$, E — заранее известный уровень энергии, собственную функцию которого мы хотим найти. Такой подход подразумевает, что спектральная задача уже решена, и поэтому не может быть использован в решении задач со сложными гамильтонианами, спектр которых не известен.

2.3 Тестовая реализация и её результаты

Попробуем найти собственные функции оператора Лапласа по известным собственным значениям. Граничные условия учтём регуляризацией волновой функции, а в качестве регуляризующей функции возьмём функцию

$$A(\mathbf{x} \in \mathbb{R}^N) = \prod_{i=1}^N \left(\left(x_i - \frac{1}{2} \right)^2 - \frac{1}{4} \right). \quad (2.12)$$

То есть вид волновой функции факторизуется

$$\psi(\mathbf{x}) = \mathcal{N}(\mathbf{x}) A(\mathbf{x}), \quad (2.13)$$

где $\mathcal{N}(\mathbf{x})$ — нейронная сеть. А в качестве функции ошибок возьмём невязку в виде (2.10).

При задании некоторого перцептрона (выбор этих параметров пока плохо изучен и производился наобум) получилось вполне успешно решить такую тестовую задачу. Реализация и её результаты для размерностей $N = 1, 2, 3$ могут быть найдены по ссылке.

2.4 Поиск основного состояния

2.4.1 Подход №1

Попробуем найти основное состояние многомерного оператора Лапласа в многомерном кубе с помощью многослойного перцептрона. Для этого снова будем учитывать граничные условия регуляризацией волновой функции, как в разделе (2.3). А в качестве функционала ошибок возьмём «усовершенствованную» невязку

$$I = \left\langle \left(\nabla^2 \psi + \frac{\langle |-\nabla^2 \psi| \rangle}{\langle |\psi| \rangle} \psi \right)^2 \right\rangle, \quad (2.14)$$

где $\langle \dots \rangle$ обозначает усреднение по всем точкам рассматриваемого набора. При этом энергия определяется как

$$E_0 = \frac{\langle |-\nabla^2 \psi| \rangle}{\langle |\psi| \rangle}. \quad (2.15)$$

При таком задании функционала ошибок удаётся найти основное состояние достаточно быстро даже в многомерном случае (см. ссылку). Форма волновой функции, видимо, определяется правильно, но энергия получается определённой не очень точно. Однако, вопрос точности определения энергии может быть легко решён, например, подходом, основанным на рассмотрении ансамбля моделей, когда создаются несколько десятков моделей и отдельно тренируются, в качестве наиболее точного значения энергии берут среднее значений, которые дают модели. Так как процесс обучения получился крайне быстрым, такой подход может быть полезен.

2.4.2 Подход №2

В данном подходе в качестве функции потерь взята такая функция:

$$I = \left\langle \left(\nabla^2 \psi + \frac{\langle -\psi \nabla^2 \psi \rangle}{\langle \psi^2 \rangle} \psi \right)^2 \right\rangle. \quad (2.16)$$

Использование такой функции потерь приводит, вроде бы, к более стабильной сходимости, но это может быть просто следствием более качественного выбора параметров нейронной сети, которая использовалась.