

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики
Кафедра моделювання складних систем

**Кваліфікаційна робота
на здобуття ступеня бакалавра**
за спеціальністю 113 Прикладна математика
на тему:

**Застосування нейронних мереж для
апроксимації функцій**

Виконала студентка 4-го курсу
Козак Дарина Дмитрівна

Науковий керівник:
професор, доктор фізико-математичних наук
Семенов Володимир Вікторович

Київ-2021

Реферат

Обсяг роботи 52 сторінки, 17 ілюстрацій, 1 таблиця, 5 джерел посилань, 4 додатки.

Ключові слова:

АПРОКСИМАЦІЯ, НЕЙРОННА МЕРЕЖА, ТЕОРЕМА СТОУНА, УНІВЕРСАЛЬНА АПРОКСИМАЦІЙНА ТЕОРЕМА, СИМПЛЕКЦІЙНІ КОМПЛЕКСИ, СИМПЛЕКЦІЙНА АПРОКСИМАЦІЙНА ТЕОРЕМА, БАГАТОШАРОВИЙ ПЕРЦЕПТРОН, РАДІАЛЬНА БАЗИСНА ФУНКЦІЯ.

Опис:

Дана робота присвячена вивченню обчислювальних можливостей нейронних мереж для задачі апроксимації функцій, обґрунтуванню методів та алгоритмів апроксимації довільних неперервних функцій за допомогою нейронних мереж, а також раціональності використання цього підходу для даного класу задач.

Зміст

Вступ	4
Задача апроксимації	6
Узагальнення теореми Стоуна	7
Нейронні мережі	8
Універсальна апроксимаційна теорема	9
Симплекційні комплекси	11
Розширення симплекційної апроксимаційної теореми	16
Мережа, що здійснює кусково-сталу апроксимацію	23
Багатошаровий перцептрон	31
Мережа радіальних базисних функцій	35
Висновки	40
Література	41
Додатки	42

Вступ

Апроксимація функцій є фундаментальною задачею у багатьох прикладних галузях, таких як аналіз даних, розпізнавання образів, прогнозування, класифікація, у тих випадках коли становить інтерес відображення між вхідними та вихідними даними, але його явна формула невідома, натомість задано лише пари $(x, f(x))$.

Існують різні методи, що вирішують цю задачу, і класичні чисельні методи зазвичай потребують підбору багатьох параметрів для апроксимації людиною, таких як вигляд апроксимуючої функції, її степінь, якщо це поліном, від яких напряду залежать результати та точність апроксимації. Окрім цього, набуває поширення використання штучних нейронних мереж, тому що в теорії це може дозволити створити алгоритм, який апроксимуватиме довільну неперервну функцію на заданому проміжку із заданою точністю, і при цьому бути повністю автоматизованим.

Основною проблемою є те, що при вивченні обчислювальних можливостей штучних нейронних мереж та їх використання з метою апроксимації функцій, виникає багато питань щодо обмеженості класів задач, які вони можуть вирішувати, точності апроксимації, необхідних умов для їх застосування, а також пошуку явного конкретного алгоритму, що дозволить цілком автоматизувати процес апроксимації довільної функції.

В роботі розглянуто теоретичні аспекти, що допомагають вирішити ці питання та можливості розширення класу задач апро-

ксімації, що можуть бути вирішені за допомогою використання нейронних мереж. Основною метою роботи є конструкція нейронної мережі, яка дозволить, обравши довільну неперервну функцію на деякому компактї, апроксимувати її із заданою точністю, що на сьогоднішній день залишається відкритим питанням.

Задача апроксимації

Загалом, апроксимація функцій дуже широко застосовується в усіх областях, де потрібне їх представлення. Отже, розглянемо задачу апроксимації, маємо:

$x_i \in \mathbb{R}^m$, $i = 1, 2, \dots, N$ - N вхідних векторів;

$d_i \in \mathbb{R}^1$, $i = 1, 2, \dots, N$ - N дійсних чисел.

Будемо шукати невідому функцію $f(x) : \mathbb{R}^m \rightarrow \mathbb{R}^1$, що задовольняє інтерполяцію, для якої:

$$f(x_i) = d_i, i = 1, 2, \dots, N.$$

Метою є мінімізації похибки апроксимації, яка найчастіше обчислюється за формулою:

$$E(f) = \frac{1}{2} \sum_{i=1}^N (d_i - f(x_i))^2.$$

Узагальнення теореми Стоуна

Базовим теоретичним результатом в області апроксимації функцій є теорема Вейерштрасса, що стверджує: неперервну функцію від n змінних - $f(x_1, x_2, \dots, x_n)$ на замкненій обмеженій множині можна рівномірно наблизити послідовністю поліномів.

Проте результат теореми Вейерштрасса можна узагальнити та посилити [6], розглянувши компактний простір X та алгебру $C(X)$ неперервних дійснозначних функцій на X , теоремою Стоуна: нехай $E \subseteq C(X)$ - замкнена підалгебра в $C(X)$, $1 \in E$, де 1 - функція, визначена рівністю $1(x) = 1, x \in X$, і для довільних $x, y \in X$ існує функція $g \in E$, така що $g(x) \neq g(y)$, тоді $E = C(X)$.

Таким чином теорема Стоуна розглядає функції на довільному компактi, і доводить щільність кільця многочленів від довільного набору функцій, що розділяють точки. Окрім апроксимації функцій многочленами та їх узагальненнями, можливо реалізувати наближення функцій багатьох змінних за допомогою композицій та лінійних операцій, саме цим оперують штучні нейронні мережі. Зазначимо, що наступні розглянуті теореми про апроксимацію, що стосуються нейронних мереж, охоплюють класичну теорему Стоуна.

Нейронні мережі

Штучні нейронні мережі, як відомо, є штучними аналогами біологічних нейронних мереж мозку людини. В загальному, нейронну мережу можна формально представити у вигляді відображення $N_{\omega, \theta} : \mathbb{R}^n \rightarrow \mathbb{R}^m$, що залежить від набору ваг ω та набору параметрів θ , до якого можна віднести вид активаційної функції, шари, зв'язки між нейронами та інші особливості архітектури.

Детальніше розглядатимемо один із найпростіших видів штучних нейронних мереж - багатошарову нейронну мережу прямого зв'язку, що складається з 3 або більше з'єднаних між собою шарів нейронів: вхідного шару, 1 або кількох прихованих шарів та вихідного шару.

Означимо формально:

Багатошаровою нейронною мережею з прямим зв'язком, визначеною на дійснозначному n -вимірному просторі, є відображення $N : \mathbb{R}^n \rightarrow \mathbb{R}^m$, таке що для кожного $x \in \mathbb{R}^n$, $N(x)$ є композицією функцій: $N(x) = f_{k+1} \circ f_k \circ \dots \circ f_1(x)$, де $k \in \mathbb{Z}$ - кількість прихованих шарів, $k \geq 1$, $f_i : \mathbb{R}^{d_{i-1}} \rightarrow \mathbb{R}^{d_i}$, $1 \leq i \leq k+1$ визначаються так: $f_i(y) = \phi_i(W^{(i)}; y; b_i)$, де $W^{(i)} = W_{d_{i-1}}^{d_i}$ - дійснозначна матриця розмірності $d_{i-1} \times d_i$, $b_i \in \mathbb{R}^{d_i}$ - величина зміщення, ϕ_i - активаційна функція (обмежена, неперервна та не тотожно стала), при цьому $d_0 = n$, $d_{k+1} = m$, $d_i \in \mathbb{Z}$, $1 \leq i \leq k$ - ширина i -того прихованого шару, $\max\{d_0, \dots, d_{k+1}\}$ - ширина мережі.

Універсальна апроксимаційна теорема

Розглянемо першу і другу апроксимаційні теореми, що є одними з перших результатів розвитку нейронних мереж, і доводять, що багат шарова нейронна мережа прямого зв'язку з одним прихованим шаром і не поліноміальною активаційною функцією, наприклад сигмоїдою:

$\sigma(x) = \frac{1}{1+e^{-x}}$, може апроксимувати будь-яку функцію, неперервну на компактній множині з \mathbb{R}^n . Проте існують недоліки, що стають на шляху практичного використання цього методу, такі як експоненціальне зростання ширини прихованого шару, та відсутність конструктивного доведення цих фактів, що не дозволяє увести однозначний алгоритм для побудови такої мережі.

Усе це робить теоретичні аспекти побудови нейронних мереж та питання щодо їх меж у можливому використанні як універсальних апроксиматорів цікавим для багатьох дослідників. Зокрема, можна показати, що існують неперервні на компактній множині функції, що не можуть бути апроксимовані нейронною мережею за умови обмеженості ширини шарів, незалежно від глибини мережі.

Будемо розглядати метод знаходження ваг багат шарової нейронної мережі прямого зв'язку з двома прихованими шарами, що апроксимує довільне скінченне відображення між двома триангульовними множинами у метричному просторі, що має конструктивне доведення, а тому залежить лише від бажаної точності апроксимації. Підхід базується на результатах обчислювальної топології: можливості моделювання триангульовних множин

симплекційними комплексами, та відповідно апроксимувати неперервне відображення між двома тріангульованими множинами відображенням між симплекційними комплексами, а також можливості моделювання відображення між симплекційними комплексами нейронними мережами прямого зв'язку з двома прихованими шарами.

Користуючись введеними позначеннями, сформулюємо один з найважливіших теоретичних результатів вивчення нейронних мереж прямого зв'язку - універсальну апроксимаційну теорему (або теорему Цибенка) [7]:

Універсальна апроксимаційна теорема: нехай A - довільна компактна підмножина \mathbb{R}^n , $C(A)$ - простір дійснозначних неперервних функцій на A , тоді для будь-якого $\epsilon > 0$ і будь-якої функції $g \in C(A)$, існує нейронна мережа з прямим зв'язком з одним прихованим шаром $N : \mathbb{R}^n \rightarrow \mathbb{R}$, визначена як $N(x) = f_2 \circ f_1(x)$, де $f_1(y) = \phi_1(W^{(1)}; y; b_1)$, $f_2(y) = W^{(2)}y$, і N апроксимує функцію g : $|N(x) - g(x)| < \epsilon$ для всіх $x \in A$.

Проте залишається відкритим питання про алгоритм побудови мережі, яка здійснюватиме апроксимацію для кожної конкретної функції, оскільки відомі доведення теореми не є конструктивними, і не пропонують загального алгоритму побудови мережі, тому розглядати їх тут не будемо. Натомість можемо запропонувати конструктивне доведення за умови, що мережа матиме 2 прихованих шари, а результат буде дійсним для тріангульованих множин, замість довільних компактних, проте насправді таке вирішення охоплює більшість можливих задач.

Симплекційні комплекси

Далі нам будуть потрібні поняття симплекційних комплексів та інші, пов'язані з ними факти.

Загалом, симплекційні комплекси є математичними об'єктами, що декомпонують топологічний простір на простіші частини, тому використовуються для представлення топологічних просторів. По суті, вони складаються з набору симплексів (n-вимірних тетраедрів). Щоб формалізувати поняття симплексу, введемо поняття афінної незалежності точок:

Афінною оболонкою підмножини A афінного або векторного простору є множина :

$$\left\{ \sum_{j=1}^i \lambda_j x_j \mid i > 0, x_j \in A, \lambda_j \in \mathbb{R}, \sum_{j=1}^i \lambda_j = 1 \right\}.$$

Елементи деякої множини є афінно незалежними, якщо жоден елемент цієї множини не належить афінній оболонці інших її елементів.

Таким чином формалізуємо поняття симплексу:

Нехай $v_0, v_1, \dots, v_i - i + 1$ афінно незалежна точка в $\mathbb{R}^n (i \leq n)$; тоді i-симплексом, $\sigma = (v_0, v_1, \dots, v_i)$, є опукла множина

$$\left\{ x \in \mathbb{R}^n \mid x = \sum_{j=0}^i \lambda_j v_j, \lambda_j \geq 0, \sum_{j=0}^i \lambda_j = 1 \right\},$$
 а точки v_0, v_1, \dots, v_i -

вершини σ , i - розмірність σ , σ' - грань σ , якщо σ' - i' -симплекс, де $i' \leq i$ та вершини σ' є вершинами σ .

Симплекційний комплекс K є скінченним набором симплексів, таким що:

1. $\sigma \in K$ і σ' є невласною підмножиною σ означає, що $\sigma' \in K$;
2. $\sigma, \mu \in K$ означає, що $\sigma \cap \mu$ - порожній або грань обох.

Введемо ще деякі позначення:

позначатимемо через $|K|$ простір, що є об'єднанням усіх симплексів симплекційного комплексу K разом із топологією Евклідового простору, у якому вони лежать;

L є підкомплексом симплекційного комплексу K , якщо $L \subseteq K$;

j -остовом є такий підкомплекс симплекційного комплексу K , що містить усі симплекси K розмірності $\leq j$ (позначається $K^{(j)}$);

симплекс $\sigma \in K$ є максимальним симплексом симплекційного комплексу K , якщо він не є гранню жодного іншого симплексу, що входить в K ;

симплекс σ_1 є сусіднім до симплексу σ_2 у симплекційному комплексі K , якщо σ_1 і σ_2 мають спільне ребро у K .

Наприклад, на наступному малюнку зображено симплекційний комплекс, на якому зокрема: (O) - 0-симплекс; (H, J) - 1-симплекс; (A, B, C) - 2-симплекс; (P, Q, R, N) - 3-симплекс; (A, B) - одна із граней симплекса (A, B, C):

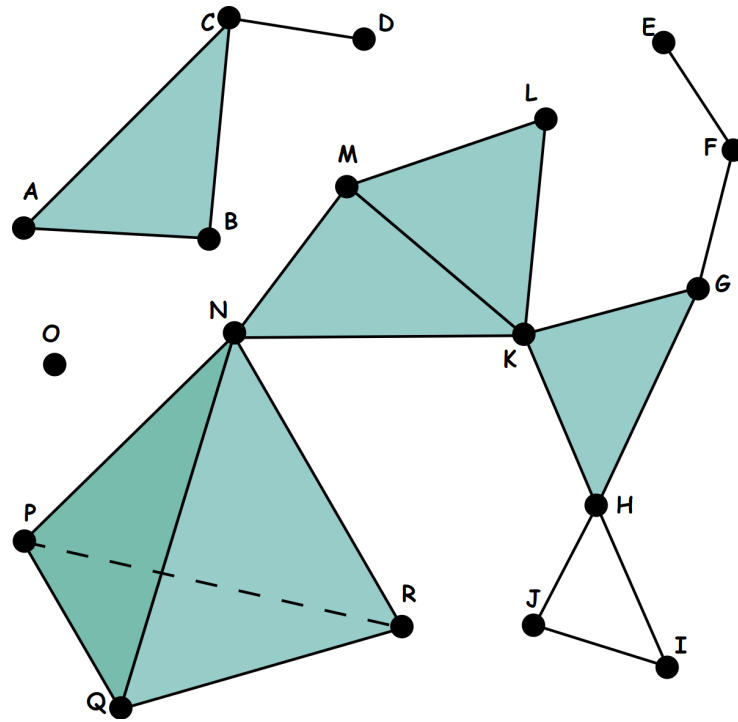
Симплекційний комплекс K' є підрозбиттям симплекційного комплексу K , якщо:

1. $|K| = |K'|$,
2. $\sigma' \in K'$ означає, що існує $\sigma \in K$, таке що $\sigma \subseteq \sigma'$.

Особливо важливим способом підрозбиття симплекційного комплексу є барицентричне підрозбиття; зазначимо, що барицентром

$$b(\sigma) \text{ і-симплексу } \sigma = (v_0, \dots, v_i) \in b(\sigma) = \sum_{j=0}^i \frac{1}{j+1} v_j.$$

Щоб ввести поняття барицентричного підрозбиття симпле-



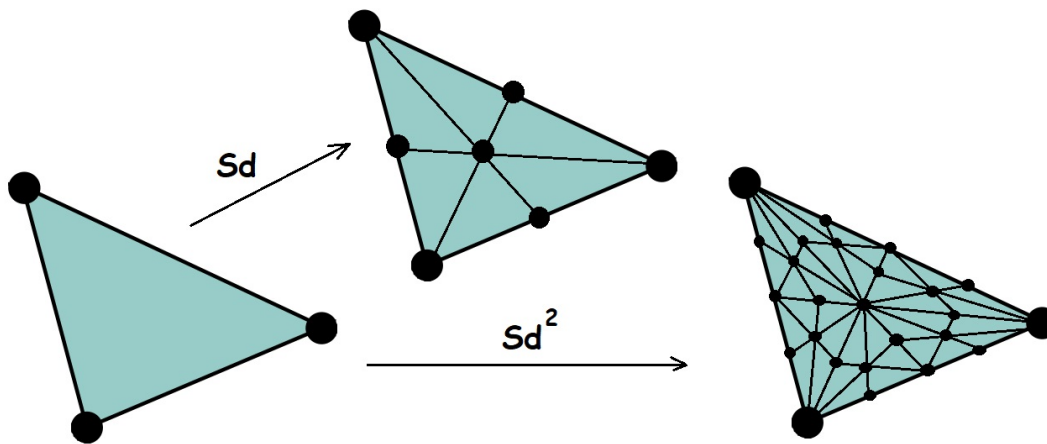
кційного комплексу (позначатимемо $Sd K$), зазначимо, що барицентричне підрозбиття 0-остова K визначається як множина вершин K , тобто $SdK^{(0)} = K^{(0)}$, і маючи $SdK^{(i-1)}$ - барицентричне підрозбиття $(i-1)$ -остова K , $SdK^{(i)}$ будується шляхом додавання барицентра кожного i -симплекса як нової вершини і з'єднання його із симплексами, що розбиває межі цього i -симплексу. Ітераційне застосування операції барицентричного підрозбиття позначається $Sd^t K$, t - кількість застосувань.

Проілюструємо на малюнку:

Діаметром симплексу σ симплекційного комплексу K (позначається $\delta(\sigma)$) є: $\delta(\sigma) = \max\{\|x - y\| \mid x, y \in \sigma\}$, а $\max\{\delta(\sigma) \mid \sigma \in K\}$ позначатимемо через $m(K)$.

Th 1: Нехай K - симплекційний комплекс, $\epsilon > 0$, тоді існує $t > 0$, таке що $m(Sd^t K) \leq \epsilon$.

Зіркою симплексу σ симплекційного комплексу K є множина усіх симплексів із K , гранню яких є σ , позначатимемо $st(\sigma)$, формально:



$$st(\sigma) = \{\mu \in K \mid \exists \xi \in K : \sigma \subseteq \xi, \mu \subseteq \xi\}.$$

Нехай K і L - симплекційні комплекси, картою вершин називається функція $\varphi : K^{(0)} \rightarrow L^{(0)}$, така що відображає вершини кожного симплексу із K у вершини симплексу із L .

Карта вершин φ може бути розширена до неперервного відображення $\varphi_c : |K| \rightarrow |L|$:

нехай K і L - симплекційні комплекси, і $\varphi : K^{(0)} \rightarrow L^{(0)}$ - карта вершин, формалізуємо поняття симплекційного відображення φ_c , породженого відображенням φ : нехай $x \in |K|$, тоді існує i -симплекс $\sigma = (v_0, \dots, v_i)$ із K і такі $\lambda_j \geq 0$, що $\sum_{j=0}^i \lambda_j = 1$ і

$$x = \sum_{j=0}^i \lambda_j v_j, \text{ тоді } \varphi_c = \sum_{j=0}^i \lambda_j \varphi(v_j).$$

При цьому якщо $g : |K| \rightarrow |L|$ - неперервне відображення, симплекційне відображення $\varphi_c : |K| \rightarrow |L|$, породжене картою вершин $\varphi : K^{(0)} \rightarrow L^{(0)}$, є симплекційною апроксимацією, якщо $g(|st(v)|) \subset |st(\varphi(v))|$ для кожної вершини v із K .

І нарешті, оперуючи усіма введеними поняттями та результатами, можемо перейти до розгляду важливого результату, що базується на теорії симплекційних комплексів - симплекційної апро-

ксимаційної теореми:

Симплекційна апроксимаційна теорема: Якщо $g : |K| \rightarrow |L|$
- неперервне, то існує достатньо велике значення $t > 0$, що $\varphi_c :$
 $|Sd^t K| \rightarrow |L|$ - симплекційна апроксимація g .

Розширення симплекційної апроксимаційної теореми

Розглядаючи симплекційну апроксимацію між двома симплекційними комплексами, зауважимо, що апроксимаційний підхід використовує карту вершин, а вона у свою чергу може бути виражена як штучна нейронна мережа. До того ж, симплекційна апроксимація може бути підібрана таким чином [2], що апроксимуватиме неперервну карту між просторами, що лежать в основі симплекційних комплексів. Параметри нейронної мережі можуть бути отримані з карти вершин, тому таким шляхом можна знайти нейронну мережу, що апроксимує неперервну функцію на просторах, що лежать в основі двох симплекційних комплексів. З цих міркувань можна зробити висновок, що симплекційна апроксимаційна теорема може бути взята за основу для отримання результату, близького до універсальної апроксимаційної теореми.

Для цього формалізуємо зв'язок між топологічними просторами та симплекційними комплексами:

Тріангуляція топологічного простору X складається з симплекційного комплексу K та гомеоморфізму $\tau : X \rightarrow |K|$.

Зазначимо, що простори, що тріангульовні симплекційними комплексами є всюди компактними і можуть бути поміщеними в \mathbb{R}^n для деякого n .

Введемо розширені поняття, пов'язані з діаметром симплексів:

Нехай X - тріангульований топологічний простір, (K, τ) - тріангуляція X , розширений діаметр симплекса визначається:

$$\tilde{\delta}(\sigma) = \max\{\|x - y\| : x = \tau^{-1}(a), y = \tau^{-1}(b), a, b \in \sigma\},$$

і $\max \{\tilde{\delta}(\sigma) | \sigma \in K\}$ позначається через $\tilde{m}_{(K,\tau)}(X)$.

Отже, в наступній теоремі розглянемо побудову штучної нейронної мережі прямого зв'язку з двома прихованими шарами для симплекційного відображення $\varphi_c : |K| \rightarrow |L|$, де усі симплекси K та L є максимальними симплексами з максимальною розмірністю.

Th 2: Нехай $\varphi_c : |K| \rightarrow |L|$ - симплекційне відображення, де K і L побудовані із максимальних симплексів максимальних розмірностей, тоді штучна нейронна мережа прямого зв'язку з двома прихованими шарами N_φ , така що $\varphi_c(x) = N_\varphi(x)$ для всіх $x \in |K|$ може бути визначена явно.

Доведення:

Вважатимемо, що $|K| \subset \mathbb{R}^n$ і $|L| \subset \mathbb{R}^m$, K складається із k максимальних n -симплексів $\{\sigma_1, \dots, \sigma_k\}$, де $\sigma_i = [v_0^i, \dots, v_n^i]$ для всіх i , L складається із l максимальних m -симплексів $\{\mu_1, \dots, \mu_l\}$, де $\mu_j = [v_0^j, \dots, v_m^j]$ для всіх j .

Визначимо багат шарову мережу прямого зв'язку N_φ такої структури:

вхідний шар складається із $d_0 = n$ нейронів;

перший прихований шар складається з $d_1 = k(n+1)$ нейронів;

другий прихований шар складається з $d_2 = l(m+1)$ нейронів;

і вихідний шар складається з $d_3 = m$ нейронів.

Тоді $N_\varphi = f_3 \circ f_2 \circ f_1$, де $f_i(y) = \phi(W^{(i)}; y; b_i)$, $i = 1, 2, 3$. Представимо точку x із \mathbb{R}^n у вигляді $k(n+1)$ вектора. Цей вектор може бути розглянутий як співставлення k векторів розмірності $n+1$, по одному для кожного з k симплексів комплексу K . Кожен вектор розмірності k представляє барицентричну координату x для відповідного симплексу. Матриця ваг для отримання виходу першого прихованого шару, $W^{(1)}$, і відповідне зміщення b_1 , можуть бути отримані з рівнянь для отримання барицентричних

координат таким шляхом:

$$W^{(1)} = \begin{pmatrix} W_1^{(1)} \\ \cdot \\ \cdot \\ \cdot \\ W_k^{(1)} \end{pmatrix} \in M_{k(n+1) \times n};$$

де $W_i^{(1)} \in M_{k(n+1) \times n}$ є:

$$\begin{pmatrix} v_0^i & \dots & v_n^i \\ 1 & \dots & 1 \end{pmatrix}^{-1} = (W_i^{(1)} \mid B_i); \text{ і } b_1 = \begin{pmatrix} B_1 \\ \cdot \\ \cdot \\ \cdot \\ B_k \end{pmatrix}.$$

Матриця ваг між першим та другим прихованим шаром задає карту вершин; таким чином K задано у першому прихованому шарі $k(n+1)$ нейронами, а L задано $l(n+1)$ нейронами, тоді матриця складається з 0-ів та 1-ць, де 1 - означає, що відповідна вершина з'єднана через карту вершин, 0 - ні. Тоді матриця $W^{(2)}$ визначається наступним шляхом: $W^{(2)} = (W_{s_1, s_2}^{(2)}) \in M_{l(m+1) \times k(n+1)}$, де $s_1 = j(r+1)$, $s_2 = i(t+1)$ для $i = 1, \dots, k; j = 1, \dots, l; t = 0, \dots, n; r = 0, \dots, m$; і $b_2 = 0$;

$$W_{s_1, s_2}^{(2)} = \begin{cases} 1 & \varphi(v_t^i) = w_r^j, \\ 0 & \text{інакше;} \end{cases}$$

Вихід другого прихованого шару може бути розглянутий як співставлення l векторів розмірностей $m+1$, по одному для кожного симплекса у комплексі L . Кожен з векторів представляє барицентричні координати $\varphi_c(x)$, що належить кожному симплексу. На наступному кроці $\phi_3(W^{(3)}; y; b_3)$ перетворює цей вектор і

дає на виході декартові координати $\varphi_c(x)$. Звідси:

$$W^{(3)} = (W_1^{(3)} \dots W_l^{(3)}) \in M_{m \times l(m+1)},$$

$$\text{де } W_j^{(3)} = (u_0^j \dots u_m^j); b_3 = 0.$$

Таким чином дані функції визначаються, для перших двох шарів:

$$\phi_i(W^{(i)}; y; b_i) = W^{(i)}y + b_i \text{ для } i = 1, 2.$$

Для 3 шару ϕ_3 визначається так:

$$\phi_3(W^{(3)}; y; b_3) = \frac{\sum_{j=1}^l z_j \psi(y_j)}{\sum_{j=1}^l y_j},$$

$$\text{де } z_j = W_j^{(3)} y_j, y = \begin{pmatrix} y_1 \\ \cdot \\ \cdot \\ \cdot \\ y_l \end{pmatrix} \in M^{l(m+1)}; y_j \in \mathbb{R}^{m+1}, \psi(y_j) = 1,$$

якщо усі координати $y_j \geq 0$, $\psi(y_j) = 0$ інакше.

Таким чином показано, що нейронна мережа прямого зв'язку з двома прихованими шарами може поводитись так само як симплекційне відображення.

Розглянемо розширення симплекційної апроксимаційної теореми, що передбачає явно заданий алгоритм для розрахунку симплекційної апроксимації якої завгодно точності для неперервної функції $g : |K| \rightarrow |L|$ між двома просторами, в яких лежать симплекційні комплекси K і L . Для цього розглянемо наступне твердження:

Th 3: Нехай $g : |K| \rightarrow |L|$ - неперервна функція між двома просторами, в яких лежать симплекційні комплекси K і L , тоді для довільного $\epsilon > 0$ існують $t_1, t_2 > 0$, такі що $\varphi_c : |Sd^{t_1}K| \rightarrow |Sd^{t_2}K|$ - є симплекційною апроксимацією g , і $\|g - \varphi_c\| \leq \epsilon$.

Доведення: Раніше у розділі, де розглядалися симплекційні комплекси, була сформульована теорема про те, що для довільного симплекційного комплексу K , та $\epsilon > 0$, існує $t > 0$, таке що $m(Sd^t K) \leq \epsilon$; тоді із симплекційної апроксимаційної теореми, існує t_1 , таке що $\varphi_c : |Sd^{t_1} K| \rightarrow |Sd^{t_2} K|$ є симплекційною апроксимацією g , крім того оскільки $m(Sd^{t_2} K) \leq \epsilon$, то $\|g - \varphi_c\| \leq \epsilon$. Звідси може бути побудований явний алгоритм для побудови такої симплекційної апроксимації.

Th 4: Нехай $g : |K| \rightarrow |L|$ - неперервна функція, для довільного $\epsilon > 0$, можна явно задати нейронну мережу N з прямим зв'язком з двома прихованими шарами, таку що $\|N - g\| \leq \epsilon$.

Доведення випливає із попередньої теореми (звідси існує симплекційна апроксимація φ_c функції g , така що $\|g - \varphi_c\| \leq \epsilon$) та теореми 2 (існує N_φ , така що $\varphi = N_\varphi$).

Попередні результати можуть бути розширені на триангульовні простори. Тоді якщо гомеоморфізми триангуляцій відомі, то розширення буде конструктивним. Тут ми введемо іншу, конструктивну версію універсальної апроксимаційної теореми, що за певних умов апроксимуватиме довільну неперервну функцію як із заданою точністю, користуючись симплекційною апроксимаційною теоремою.

Th 5: Нехай (K, τ) - триангуляція простору X , тоді для довільного $\epsilon > 0$ існують n та $\gamma > 0$, що якщо $m(Sd^n K) \leq \gamma$, то $\tilde{m}_{(Sd^n K, \tau_K)}(X) \leq \epsilon$.

Доведення: Нехай $x, y_0 \in |K|$, якщо x та y_0 належать $\sigma_0 \in K$, тоді $\|x - y_0\| \leq \delta(\sigma_0)$ і $\|\tau(x) - \tau(y_0)\| \leq \tilde{\delta}(\sigma_0)$. Інакше x та y_0 можуть належати до одного симплекса в SdK , або ні. Якщо так,

нехай $y_1 = y_0$; якщо ні, нехай y_1 - така що, $x, y_1 \in \sigma_1$ і $\sigma_1 \in Sd\sigma_0$. Тому $\|x - y_1\| \leq \delta(\sigma_1) \leq \delta(\sigma_0)$, і $\|\tau(x) - \tau(y_1)\| \leq \tilde{\delta}(\sigma_1) \leq \tilde{\delta}(\sigma_0)$. Можна також записати: $\|x - y_t\| \leq \delta(\sigma_n) \leq \dots \leq \delta(\sigma_1) \leq \delta(\sigma_0)$, і $\|\tau(x) - \tau(y_t)\| \leq \tilde{\delta}(\sigma_t) \leq \dots \leq \tilde{\delta}(\sigma_1) \leq \tilde{\delta}(\sigma_0)$. Таким чином задано послідовність $\{y_i\}_{i=0}^t$, що збігається до x . Тоді для довільного $\epsilon > 0$, існує n , таке що $\|\tau(x) - \tau(y_n)\| \leq \epsilon$. Без обмеження загальності вважатимемо, що $\tilde{\delta}(\sigma_n) = \tilde{m}_{(Sd^n K, \tau)}(X)$, тоді можемо вважати, що $\gamma = m(Sd_n K)$.

Наслідок: Для довільного $\epsilon > 0$ та (K, τ) - триангуляції X , існує t , таке що $\tilde{m}(Sd^t K) \leq \epsilon$.

Доведення: за теоремою 1 існує t' , таке що $m(Sd^{t'} K) \leq \gamma$, тоді із попередньої теореми існує t , таке що $\tilde{m}_{(Sd^t K, X)} \leq \epsilon$.

Нарешті, маючи неперервне відображення g між двома триангульовними просторами X та Y , можемо отримати два симплекційні комплекси K та L , що лежать в цих просторах, і симплекційну апроксимацію φ_c , що апроксимує g .

Th 6: Нехай X та Y - триангульовні топологічні простори, $g : X \rightarrow Y$ - неперервне відображення, $\epsilon > 0$, тоді існують триангуляції $(K, \tau_K), (L, \tau_L)$ просторів X, Y відповідно, і симплекційна апроксимація $\varphi_c : |Sd^{t_1} K| \rightarrow |Sd^{t_2} L|$, така що $\|g - \tau_L^{-1} \circ \varphi_c \circ \tau_K\| \leq \epsilon$.

Доведення: За останнім наслідком, існує t_2 , таке що $\tilde{m}_{(Sd^{t_2} \tilde{L}, \tau_L)}(Y) \leq \epsilon$. Далі згідно симплекційної апроксимаційної теореми: існують t_1 і $\tilde{\varphi}$, такі що $\tilde{\varphi} : Sd^{t_1} \tilde{K} \rightarrow Sd^{t_2} \tilde{L}$ є симплекційною апроксимацією $\tau_{\tilde{L}} \circ g \circ \tau_{\tilde{K}}^{-1}$. Зважаючи на те, що $|Sd^{t_1} \tilde{K}| = |\tilde{K}|$ і $|Sd^{t_2} \tilde{L}| = |\tilde{L}|$, можна визначити симплекційне відображення $\tilde{\varphi}_c : |\tilde{K}| \rightarrow |\tilde{L}|$, породжене $\tilde{\varphi}$, а також $\varphi_c = \tau_{\tilde{L}}^{-1} \circ \tilde{\varphi}_c \circ \tau_{\tilde{K}}$, $K = Sd^{t_1} \tilde{K}, L = Sd^{t_2} \tilde{L}$. Оскільки $\tilde{m}_{(L, \tau_L)}(Y) \leq \epsilon$, тоді $\|g - \tau_L^{-1} \circ \varphi_c \circ \tau_K\| \leq \epsilon$.

Th 7: Нехай $g : X \rightarrow Y$ - неперервна функція, $(K, \tau_K), (L, \tau_L)$

- триангуляції просторів X та Y відповідно. Тоді нейронна мережа N прямого зв'язку з двома прихованими шарами, що апроксимує g із заданою точністю може бути задана явно.

Доведення: Із останньої теореми, існує симплекційна апроксимація φ_c функції g , що апроксимує g з довільною точністю, і з теореми 2 існує N , що $N = \varphi_c$ на всьому інтервалі, тому N апроксимує g з довільною точністю.

Таким шляхом уведено ефективний метод побудови нейронної мережі прямого зв'язку з двома прихованими шарами, що апроксимує неперервну функцію між триангульовними просторами. Недоліками підходу може бути складність знаходження гомеоморфізмів між триангульовними просторами. Проте, основним внеском та перевагою такого підходу є те, що ваги такої мережі можуть бути зручно обраховані без процесу навчання, що дуже оптимізує розв'язання задачі.

Мережа, що здійснює кусково-сталу апроксимацію

Для прикладу практичного застосування розглянемо явну конструкцію нейронної мережі з прямим зв'язком, що здійснює кусково-сталу апроксимацію будь-якої функції однієї чи багатьох змінних [3], ваги якої визначаються явно, тому вона не потребує чисельної оптимізації для навчання. Дана конструкція передбачає використання 2 прихованих шарів зі щонайбільше n^2 нейронами. Не виключається можливість існування схожих нейронних мереж з прямим зв'язком з меншою кількістю нейронів, проте з більш складною конструкцією, це питання залишається цікавим для дослідження.

Запропонована нейромережа, окрім того, що являє собою черговий новий приклад конструктивного доведення універсальних апроксимаційних властивостей нейронних мереж з прямим зв'язком, також є практичним інструментом для дослідження реальних даних.

Постановка задачі

Розглянемо задачу апроксимації невідомої функції $f : D \rightarrow \mathbb{R}$, де D - невласна підмножина \mathbb{R}^d , $d \geq 1$,

$x = (x_1, \dots, x_d)$ - координата,

набір даних: $(x^{(1)}, f^{(1)}), \dots, (x^{(n)}, f^{(n)})$, $n > 1$,

$x^{(k)}$ - k -та координата;

$f^{(k)} = f(x^{(k)}) + \epsilon_k$,

$k=1, \dots, n$, де $\epsilon_k \geq 0$ - можлива похибка.

Вважаємо, що випадок нетривіальний, тобто $n > 1$.

Структура мережі

Мережа складається з шару входу, що містить $d \geq 1$ нейронів, які відповідають вхідному сигналу розмірності d , і шару виходу, що містить 1 нейрон, який відповідає шуканому значенню функції f . Функцією активації є функція Хевісайда (одичний стрибок):

$$s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$$

Перший прихований шар складається з $n(n-1)$ нейронів ($n > 1$). Поділимо ці нейрони на n умовних груп, кожна з яких відповідно містить $n-1$ нейронів. Для позначення приналежності нейрона першому прихованому шару використаємо позначення $N^{(1)}$ і використовуватимемо подвійну індексацію таким способом: $N_{k,j}^{(1)}$, де $1 \leq k \leq n$, $1 \leq j \leq n$, $j \neq k$. Тут індекс $k = 1, \dots, n$ вказує на номер групи, індекс j вказує на розташування у k -й групі.

Варто зауважити, що це лише схема індексації для однозначної ідентифікації нейронів надалі, при цьому усі нейрони першого прихованого шару отримують однакові сигнали із вхідного шару.

Вихід нейронів першого прихованого шару явно визначається формулою:

$$\begin{aligned} z_{k,j}^{(1)}(x) &= s(w_{k,j}x - b_{k,j}), \\ \text{де } 1 \leq k \leq n, 1 \leq j \leq n, j \neq k, \\ w_{k,j} &= x^{(k)} - x^{(j)}, \\ b_{k,j} &= \frac{1}{2}(x^{(k)} - x^{(j)})(x^{(k)} + x^{(j)}). \end{aligned}$$

Зважаючи на вигляд функції активації, вираз $z_{k,j}^{(1)}(x)$ також приймає значення з множини $\{0,1\}$. Тоді вектор виходу k -ї групи

можна записати так:

$$z_k^{(1)}(x) = (\dots, z_{k,k-1}^{(1)}, z_{k+1}^{(1)}, \dots)^T \in \{0, 1\}^{n-1},$$

$$1 \leq k \leq n.$$

Другий прихований шар складається з n нейронів, кожен з яких отримує сигнал лише з нейронів однієї відповідної йому групи першого прихованого шару. Вихід нейронів другого прихованого шару визначається формулою:

$$z_k^{(2)}(x) = s(\mathbf{1}z_k^{(1)}(x) - b^{(2)}),$$

$$k=1, \dots, n,$$

$$\mathbf{1} = (1, \dots, 1)^T \in \mathbb{R}^{n-1},$$

$$b^{(2)} = n - 1.$$

Для уникнення неочікуваних ефектів, спричинених заокруглювальною помилкою комп'ютера, при реалізації можна вважати: $b^{(2)} = n - 1 - \epsilon$, де $0 < \epsilon < 1$. Відповідно, $z_k^{(2)}(x)$ також приймає значення із множини $\{0, 1\}$.

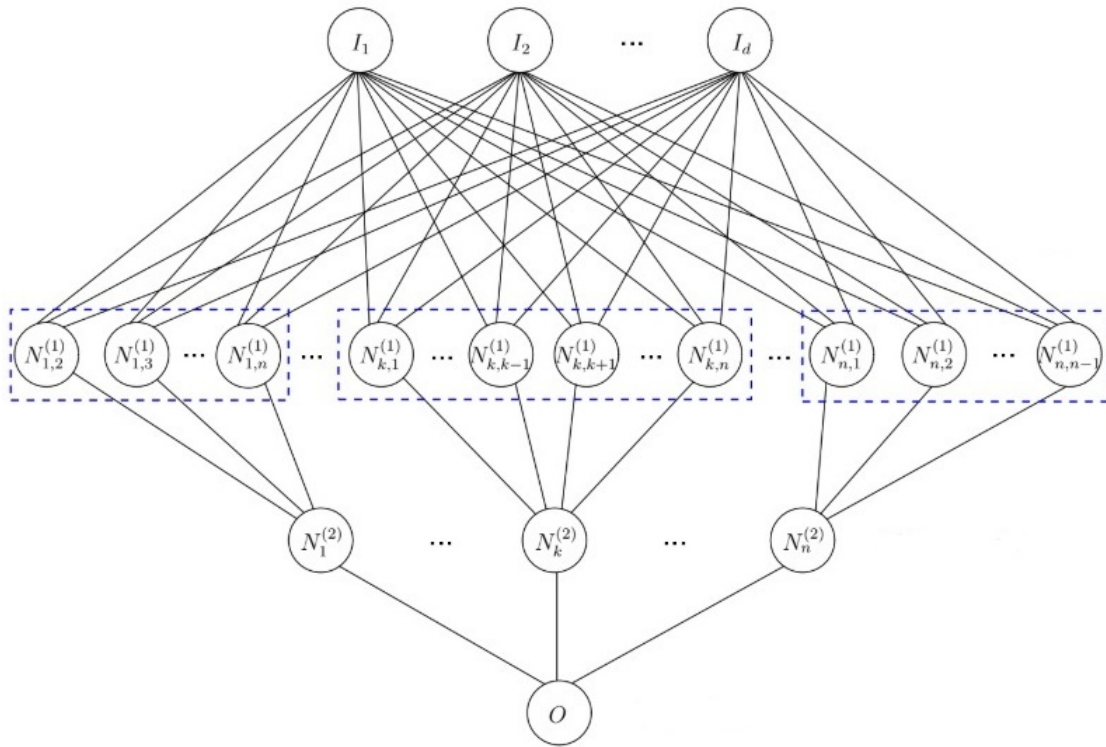
Шар виходу отримує сигнали з усіх нейронів другого прихованого шару, і вихід задається формулою:

$$y(x) = \sum_{k=1}^n f^{(k)} z_k^{(2)}.$$

Наступна схема ілюструє описану структуру:

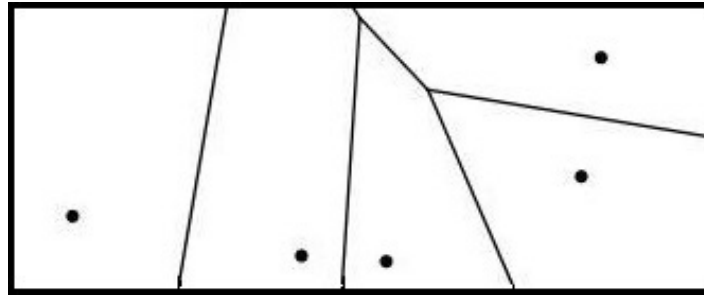
Апроксимаційні властивості

Введемо поняття діаграми Вороного для множини точок $X = \{x^{(1)}, \dots, x^{(n)}\} \subset D$, де D - невласна підмножина \mathbb{R}^d , $2 \leq n < \infty$ і $x^{(i)} \neq x^{(j)}$ для $i \neq j$. Діаграма Вороного являє собою таке розбиття множини X на області - комірки Вороного, кожна з яких містить одну із точок множини X і усі точки простору, що ближчі до даної точки множини X , ніж до будь-якої іншої точки цієї множини. Взявши за відстань між точками евклідову норму, отримаємо такі формальні означення: область



$V^{(i)} = \{x \mid \|x - x^{(i)}\|_2 \leq \|x - x^{(j)}\|_2, \forall i \neq j\}$ є коміркою Вороного точки $x^{(i)}$, а множина $V(X) = \{V^{(1)}, \dots, V^{(n)}\}$ є діаграмою Вороного для множини точок X .

Приклад діаграми Вороного на площині для 5 точок:



Сформулюємо та доведемо теорему про апроксимаційні властивості описаної нейронної мережі з прямим зв'язком:

Th: Вихід даної нейронної мережі є кусково-сталою апроксимацією функції $f(x)$ за набором даних $\{x^{(k)}, f(x^{(k)})\}_{k=1}^n$, і при $n \geq 2$:

$$y(x) = \sum_{k=1}^n f(x^{(k)}) \mathbf{I}_{V^{(k)}}(x).$$

* Тут $V^{(k)}$ - комірка Вороного точки $x^{(k)}$, відповідно індикатор:

$$\mathbf{I}_{V^{(k)}}(x) = \begin{cases} 1 & x \in V^{(k)} \\ 0 & x \notin V^{(k)} \end{cases}$$

Доведення:

Нехай $x_{k,j}^{cep} = \frac{1}{2}(x^{(k)} + x^{(j)})$ - середня точка між точками $x^{(k)}$ та $x^{(j)}$ для всіх $x^{(k)}$ та усіх $x^{(j)}$, де $k = 1, \dots, n$, $n \geq 2$, $j \neq k$. Тоді з визначення виходу першого прихованого шару $z_{k,j}^{(1)} = 1$ тоді і тільки тоді, коли $(x^{(k)} - x^{(j)})x \geq (x^{(k)} - x^{(j)})x_{k,j}^{cep}$, звідси $(x - x_{k,j}^{cep})(x^{(k)} - x^{(j)}) \geq 0$.

$\{x \mid (x - x_{k,j}^{cep})(x^{(k)} - x^{(j)}) = 0\}$ - центральна гіперплощина, що розділяє точки $x^{(k)}$ та $x^{(j)}$. Тому $(x - x_{k,j}^{cep})(x^{(k)} - x^{(j)}) \geq 0$ включає усі точки, що лежать ближче до $x^{(k)}$, аніж до $x^{(j)}$. Тому отримуємо, що:

$\{x \mid z_{k,j}^{(1)}(x) = 1\} = \{x \mid \|x - x^{(k)}\|_2 \leq \|x - x^{(j)}\|_2\}$ для всіх $1 \leq k \leq n$, $j \neq k$.

Вихід k -того нейрону другого прихованого шару $z_k^{(2)} = 1$ тоді і тільки тоді, коли $z_{k,j}^{(1)} = 1 \forall j \neq k$. Звідси:

$\{x \mid z_{k,j}^{(2)}(x) = 1\} = \{x \mid \|x - x^{(k)}\|_2 \leq \|x - x^{(j)}\|_2 \forall j \neq k\} = V^{(k)}, ix^{(k)}$.

Таким чином вихід мережі $y(x) = \sum_{k=1}^n f^{(k)} z_k^{(2)}$ є кусково-сталою функцією у формі $y(x) = \sum_{k=1}^n f(x^{(k)}) \mathbf{I}_{V^{(k)}}(x)$.

Th: Вважаючи $f(x)$ диференційовною функцією, що має обмежені похідні першого порядку, і міру μ такою, що:

$$\int_D \|x\|_2^2 d\mu(x) < \infty,$$

похибка апроксимації розглянутої мережі задовольняє нерівність:

$$\|y - f\|_{L_{d\mu}^2(D)} \leq (\sup_{x \in D} \|\nabla f(x)\|_2) \left(\sum_{i=1}^n \int_{V^{(i)}} \|x^{(i)} - x\|_2^2 d\mu(x) \right)^{\frac{1}{2}}.$$

* Тут без обмеження загальності вважаємо:

$$\int_D d\mu(x) = 1.$$

Доведення:

Із попередньої теореми:

$$\begin{aligned} \|y - f\|_{L_{d\mu}^2(D)} &= \sum_{i=1}^n \int_{V^{(i)}} |f(x^{(i)}) - f(x)|^2 d\mu(x) = \\ &= \sum_{i=1}^n \int_{V^{(i)}} \left| \int_0^1 (x^{(i)} - x) \nabla f(tx^{(i)} + (1-t)x) dt \right|^2 d\mu(x) \leq \\ &= \sum_{i=1}^n \int_{V^{(i)}} (\|x^{(i)} - x\|_2 \int_0^1 \|\nabla f(tx^{(i)} + (1-t)x)\|_2 dt)^2 d\mu(x) \leq \\ &= \sum_{i=1}^n (\sup_{x \in V^{(i)}} \|\nabla f(x)\|_2)^2 \int_{V^{(i)}} \|x^{(i)} - x\|_2^2 d\mu(x) = \\ &= (\sup_{x \in D} \|\nabla f(x)\|_2)^2 \sum_{i=1}^n \int_{V^{(i)}} \|x^{(i)} - x\|_2^2 d\mu(x). \end{aligned}$$

Наслідок: Вважаючи D обмеженою:

$$\|y - f\|_{L_{d\mu}^2(D)} \leq (\sup_{x \in D} \|\nabla f(x)\|_2) \delta,$$

де $\delta = \max_{1 \leq i \leq n} \text{diam}(V^{(i)})$,

$\text{diam}(A) = \sup_{x, x' \in A} \|x - x'\|_2$ - діаметр обмеженого набору A .

Доведення випливає із попередньої теореми, бо:

$$\sum_{i=1}^n \int_{V^{(i)}} \|x^{(i)} - x\|_2^2 d\mu(x) \leq \delta^2 \sum_{i=1}^n \int_{V^{(i)}} d\mu(x) = \delta^2.$$

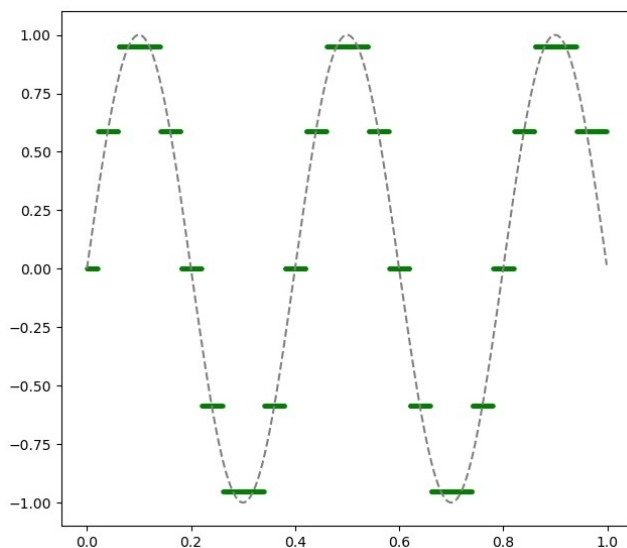
При цьому якщо вхідні дані $\{x^{(j)}\}_{j=1}^n$ нормально розподілені на обмеженій множині D , то $\delta \sim n^{-\frac{1}{d}}$, де d - розмірність простору, звідси похибка мережі виражається так:

$$\|y - f\|_{L^2_{d\mu}(D)} \sim O(n^{-\frac{1}{d}}).$$

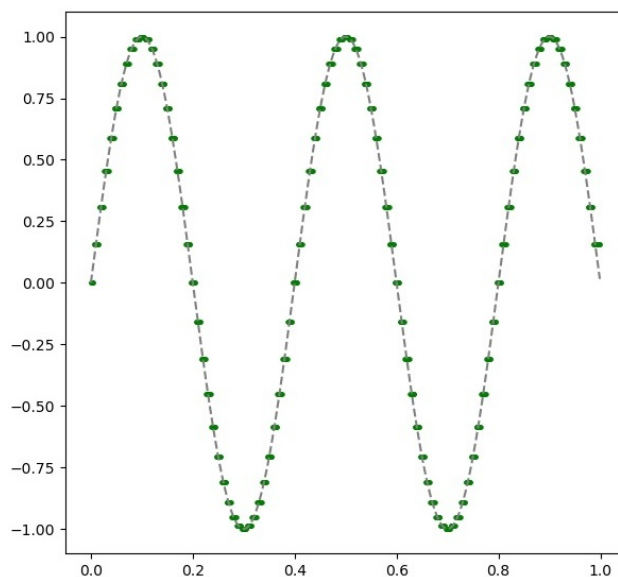
Обчислювальні експерименти

Програмна реалізація запропонованої мережі для випадку однієї змінної (для функції $\sin(5\pi x)$) з рівномірно розподіленими на відрізку $[0, 1]$ навчальними даними мовою Python наведена у додатку А.

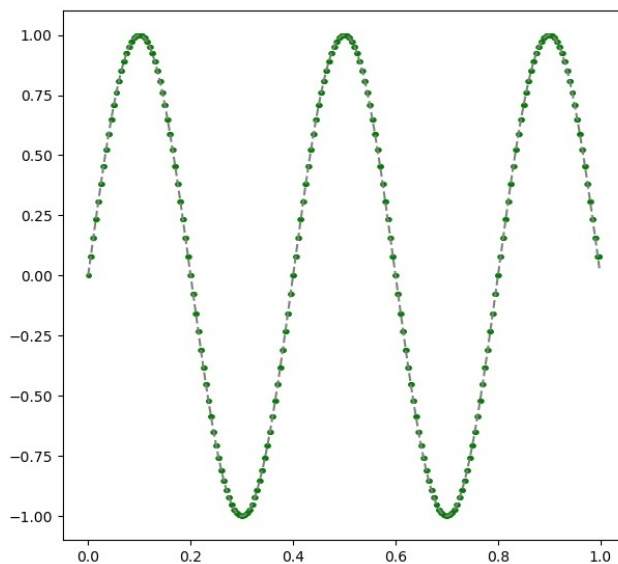
Будемо досліджувати 1000 вхідних даних. Візьмемо спочатку невеликий набір рівномірно розподілених навчальних даних - 25:



100:



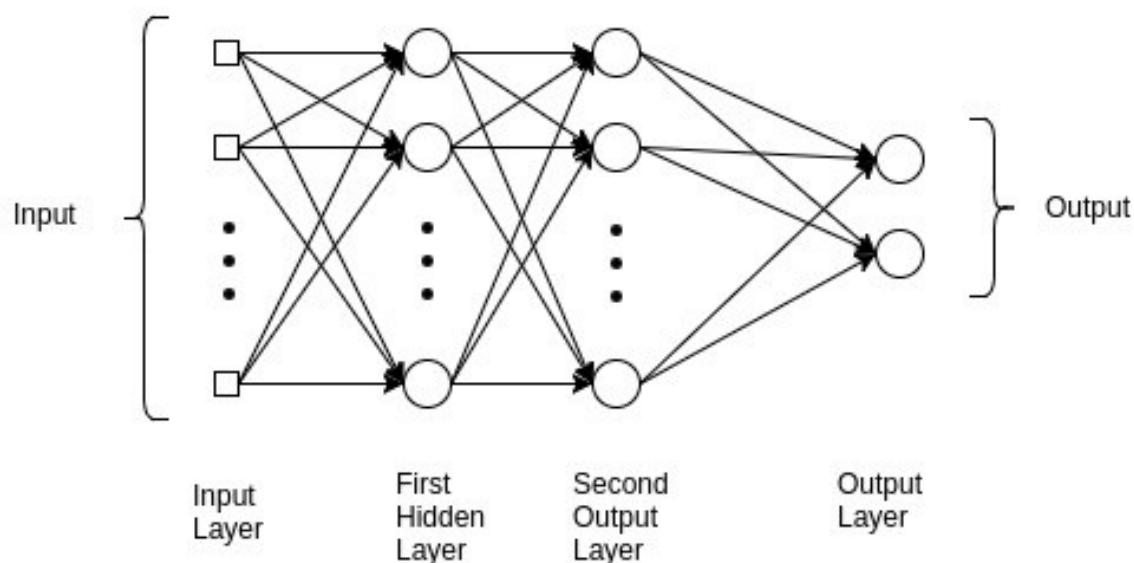
200:



З графіків бачимо, що отримано кусково-сталу апроксимацію досліджуваної функції, при цьому з ілюстрацій чітко видно, що зі зростанням кількості навчальних даних, зростає і точність апроксимації. Отже, така апроксимація є досить ефективною і даний алгоритм буде її із максимальною можливою точністю і не потребує довгого часу виконання, проте кусково-стала апроксимація сама по собі є не найефективнішим методом, тому далі ми спробуємо реалізувати більш складні конструкції.

Багатошаровий перцептрон

Тепер, згідно Універсальної апроксимаційної теореми, реалізуємо багатошаровий перцептрон з двома прихованими шарами для нашої задачі, його загальний схематичний вигляд:



Будемо використовувати бібліотеку TensorFlow. Для апроксимації оберемо функцію $\sin(x)$ на проміжку $[0,10]$. Згенеруємо 10000 випадково розподілених навчальних точок і 500 тестових. Функцією активації буде функція ReLU, що має вигляд:

$$r(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases}$$

Реалізація наведена у **додатку Б**.

Отримаємо таку модель:

Model: "sequential_1"

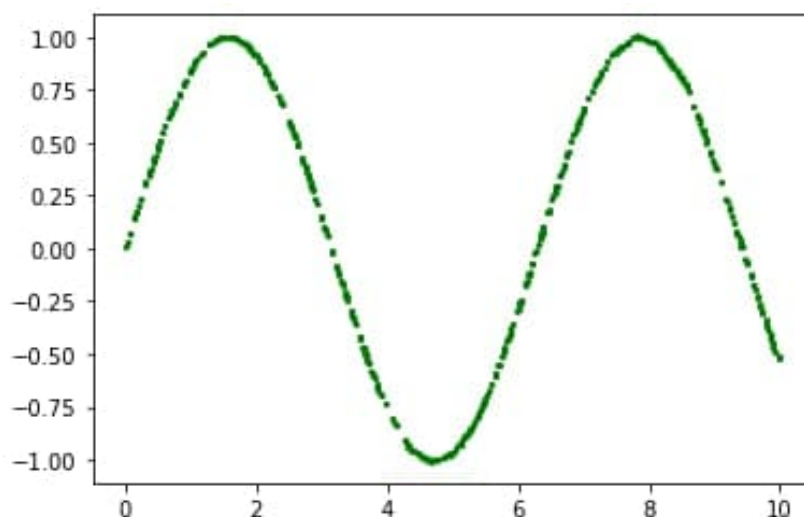
Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 500)	1000
dense_2 (Dense)	(None, 100)	50100
dense_3 (Dense)	(None, 1)	101

Total params: 51,201

Trainable params: 51,201

Non-trainable params: 0

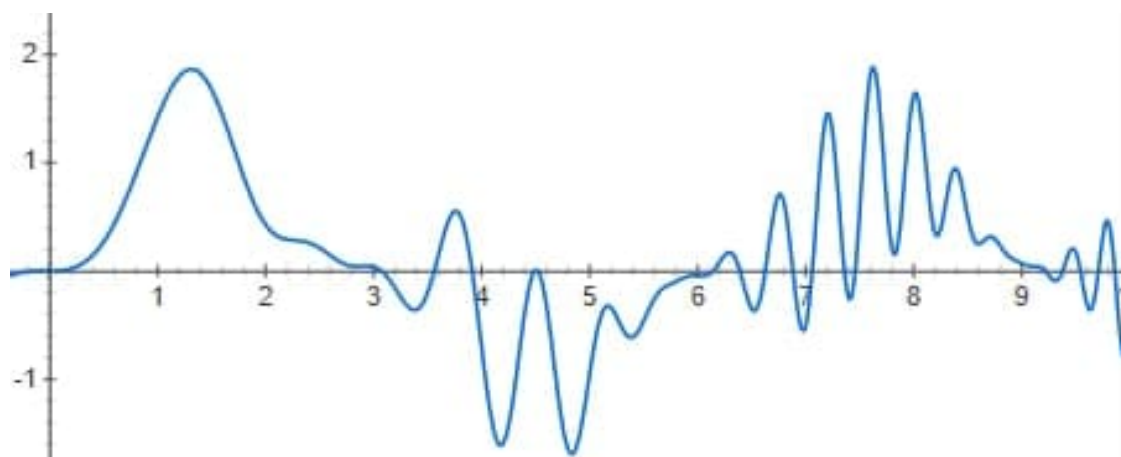
Оберемо оптимізатор Adam, що використовує метод стохастичного градієнтного спуску, що базується на адаптивному оцінюванні моментів першого та другого порядку, і будемо використовувати 20 епох для навчання. Натренувавши модель, отримаємо апроксимацію для тестових точок:



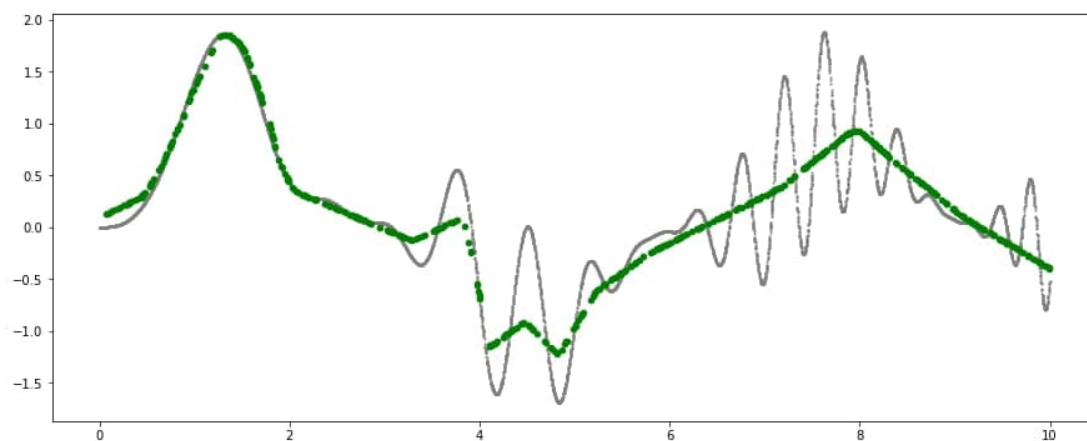
Середньоквадратична похибка складає $2.532e-05$, тому результати дуже точні.

Тепер оберемо більш складну функцію:

$y = \sin(x^2)(\cos(x + 2))^2 + (\sin(x))^3$ на проміжку $[0, 10]$, її графік:



І використаємо ту ж саму кількість даних, і ту ж мережу з тими ж параметрами, отримаємо (апроксимовані значення зеленим кольором):



Результат значно гірший, середньоквадратична похибка складає 0.11145571385591223 .

При цьому при збільшенні кількості навчальних даних, або кількості нейронів у прихованих шарах, достатньої точності досягти все одно складно, а час виконання росте настільки, що постає питання про доцільність та оптимальність використання мережі такої структури для даного класу функцій.

Проблема у тому, що використання багат шарового перцептрона має свої недоліки [4], зокрема погану збіжність для значних нелінійностей, що ми і бачимо на останньому прикладі. Крім цього, схильність застрягати на небажаних локальних мінімумах і довгий час навчання через повільну збіжність.

Тому далі, щоб оптимізувати вирішення цієї проблеми, ми розглянемо інші конструкції нейронних мереж прямого зв'язку.

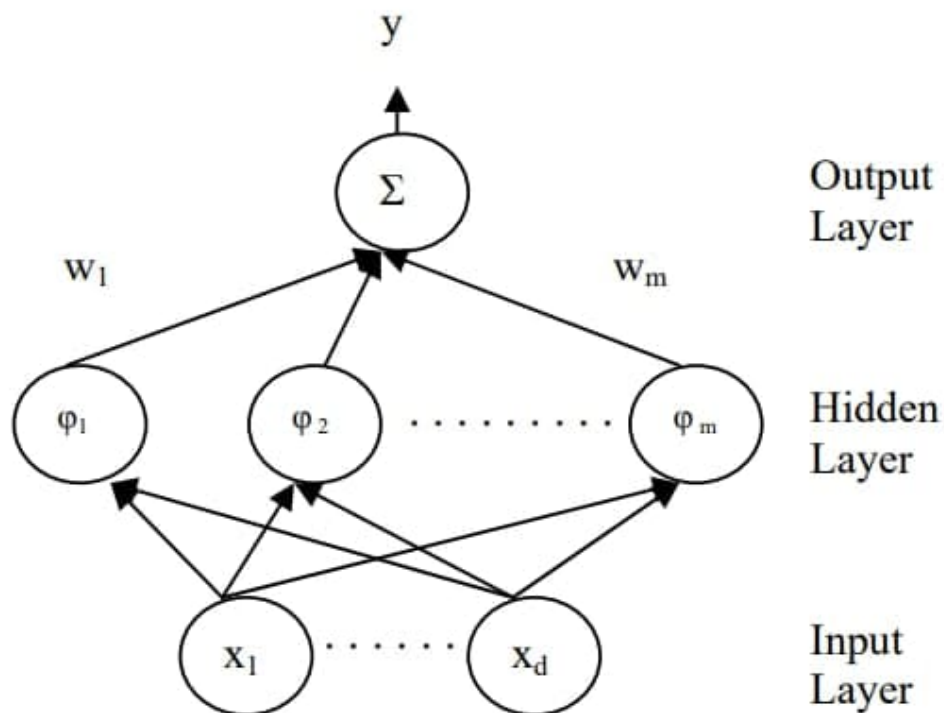
Мережа радіальних базисних функцій

Введемо поняття радіальної базисної функції:

нехай деяка точка c є центром; дійснозначна функція, чие значення залежить від відстані до центру c : $\phi(x) = \phi(\|x - c\|)$ називається радіальною базисною функцією.

Розглянемо нейронну мережу радіальних базисних функцій (RBFN - radial basis function network), яка вводиться заміною глобальної функції активації у багатошаровому перцептроні, на локальну радіальну базисну функцію [5].

Вперше така мережа була запропонована у 1988 році, і фактично, являє собою об'єднання радіальних функцій у нейронну мережу з одним прихованим шаром, схему якої можна побачити на малюнку:



До кожного прихованого вузла застосовується активаційна функція ϕ . Таким чином, вихід визначається центром \vec{c} , а також параметром b , що називається шириною:

$$\phi_j(\xi) = \phi_j(\|\vec{x} - \vec{c}\|/b), \quad j = 1, 2, \dots, m.$$

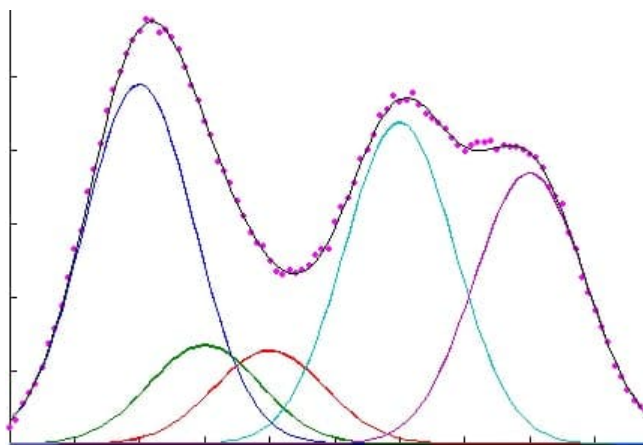
ϕ може бути довільною радіальною базисною функцією, найпоширенішими є гаусівська, мультіквадратична і обернена мультіквадратична. Центральні вектори, параметр b і вихідні ваги регулюються адаптивно у процесі навчання. Компоненти c_i на початку можуть обиратись випадково із вхідних даних, або як середнє за кластерами. За фіксованими центрами і шириною b , ваги w_i , що мінімізують похибку, обчислюються шляхом розв'язку псевдооберненої системи лінійних рівнянь [4], у матричному записі:

$$W = \Phi^+ Y,$$

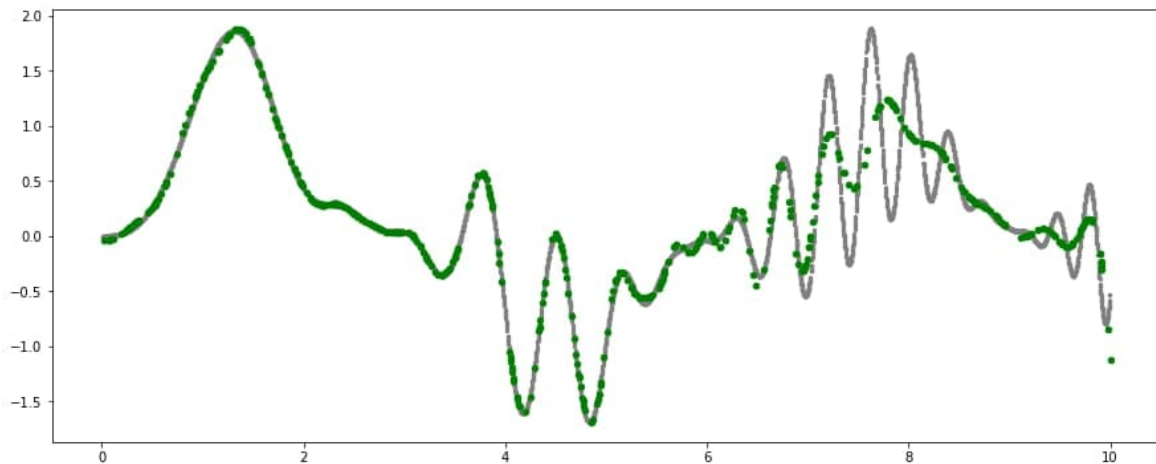
$$\text{де } \Phi = \begin{pmatrix} \phi(x_1, c_1) & \dots & \phi(x_1, c_m) \\ \phi(x_2, c_1) & \dots & \phi(x_2, c_m) \\ \dots & \dots & \dots \\ \phi(x_d, c_1) & \dots & \phi(x_d, c_m) \end{pmatrix} - \text{вихід базисних функцій } \phi,$$

і $\Phi^+ = (\Phi^T \Phi)^{-1} \Phi^T$ - псевдообернена матриця.

Тому (у випадку гаусівської активаційної функції) фактично на виході отримаємо лінійну комбінацію деяких, оптимально підібраних гаусівських функцій, як це показано на графіку:

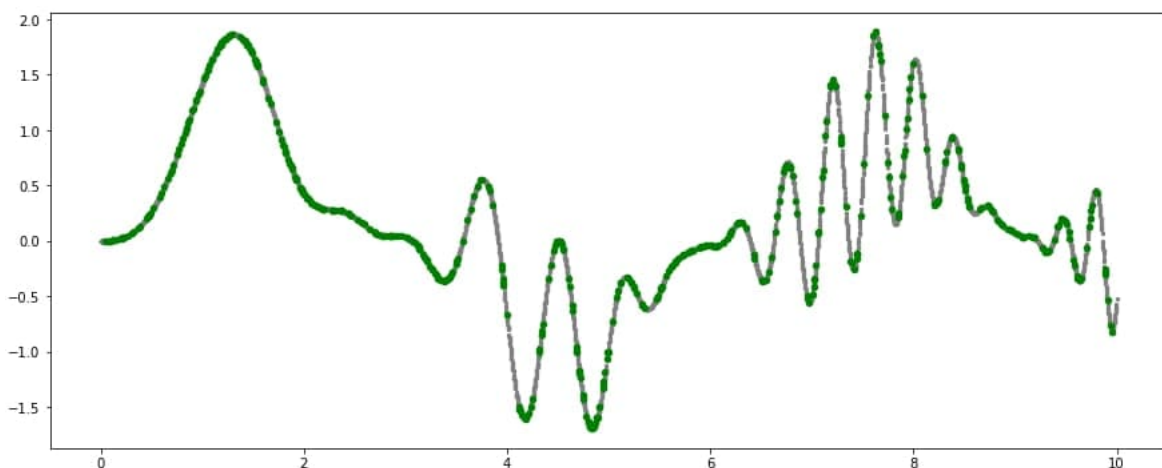


У Додатку В ралізуємо цю мережу згідно описаної структури, обравши у якості активаційної функції гаусівську. Залишається обрати кількість центрів c_i , очевидно, що чим більш складна функція, тим більше центрів знадобиться. Спробуємо апроксимувати останню функцію, описану в попередньому розділі з тою ж кількістю тренувальних та тестових точок, з кількістю центрів - 50:



Отримана апроксимація значно краща, ніж у випадку багатшарового перцептрона, середньоквадратична похибка складає 0.046879.

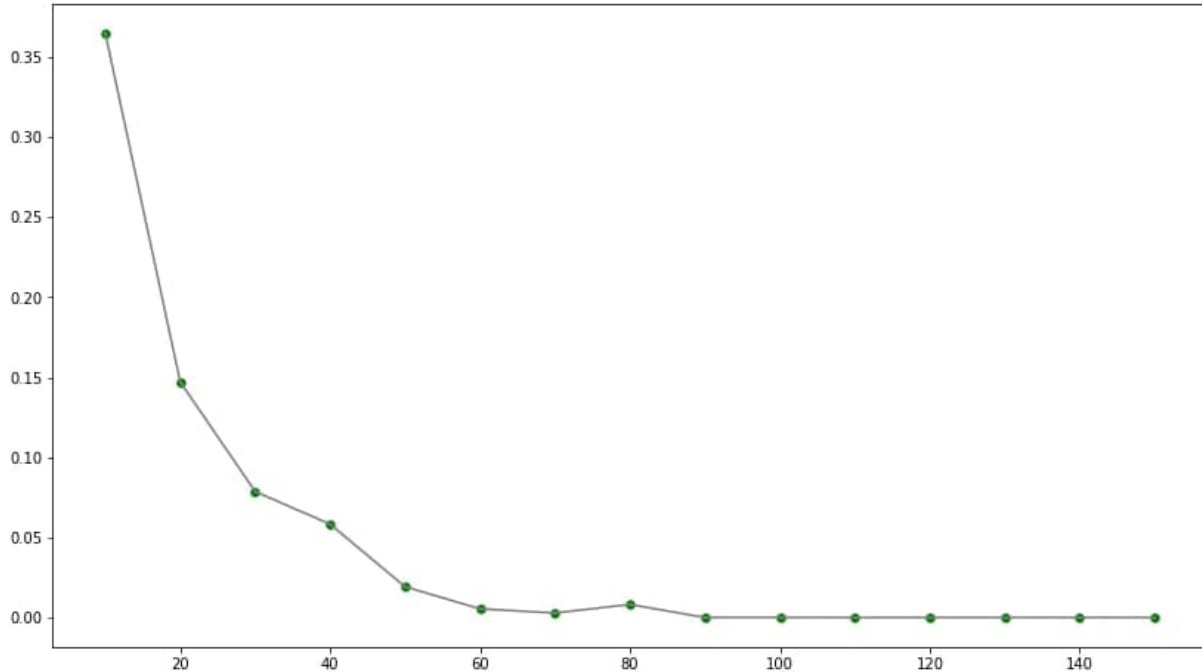
І уже збільшивши кількість центрів до 100, отримаємо:



Середньоквадратична похибка - 6.073615e-06.

При цьому час виконання - близько 12 секунд, значно менший ніж при використанні багат шарового перцептрона.

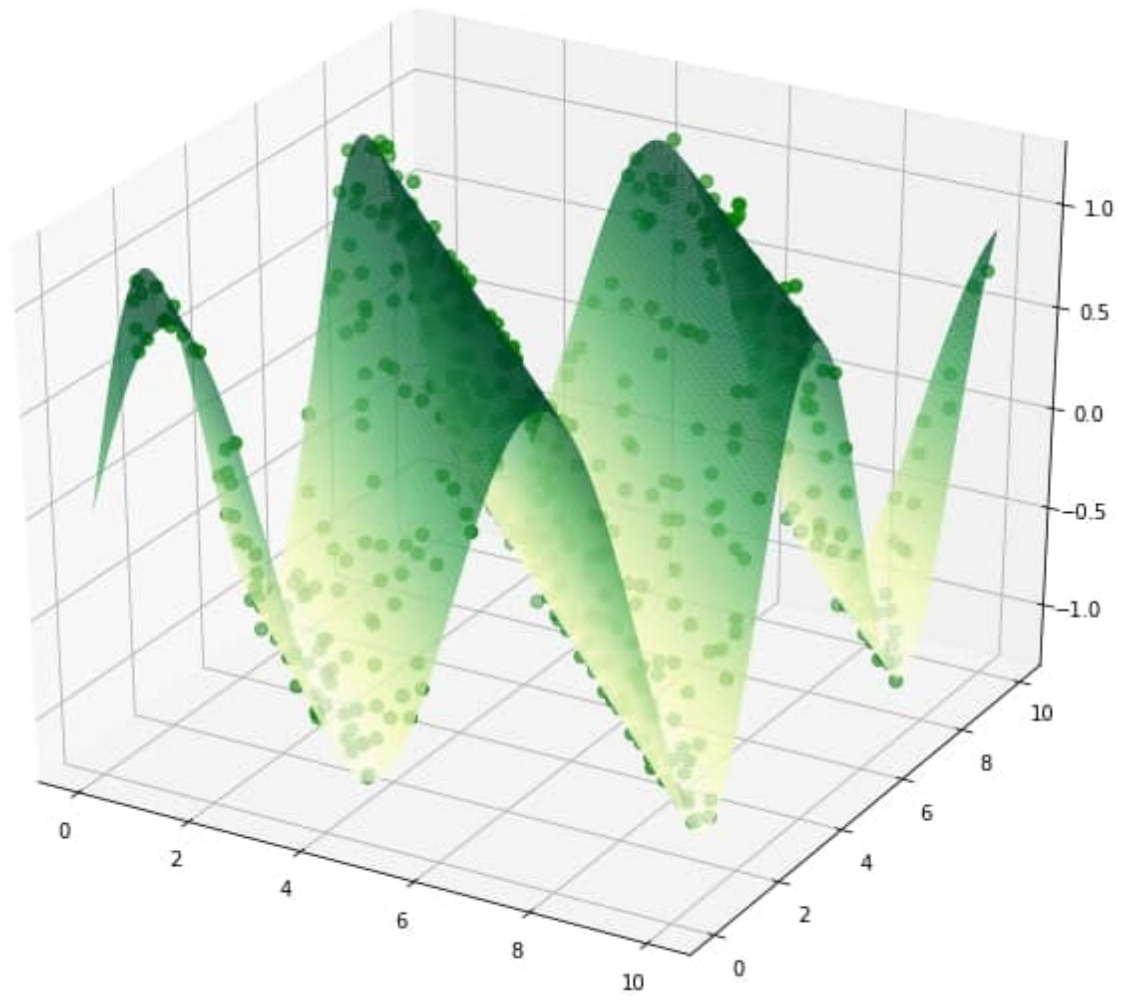
Проаналізуємо залежність похибок від кількості центрів, для кількості центрів від 10 до 150 із кроком 10:



При цьому точність, окрім кількості тренувальних даних, напряму залежить лише від кількості центрів. Отже, можна зробити висновок, що для довільно складної неперервної функції однієї змінної знайдеться достатня кількість центрів, щоб мережа радіально базисних функцій апроксимувала її із довільною точністю.

У **Додатку Г** реалізуємо описану мережу для випадку апроксимації функції багатьох змінних на прикладі функції

$z = \sin(x + y)$, при $x, y \in [0, 10]$. Зауважимо, що апроксимація функції двох змінних потребує більше часу, і точність у загальному випадку, росте повільніше, ніж для функцій однієї змінної. Використавши 1000 центрів, отримуємо апроксимацію:



Похибка складає 0.002579288, результат досить непоганий, і, аналогічно, теоретично може бути підвищений за рахунок збільшення лише кількості центрів або тренувальних точок.

Висновки

Отже, в роботі ми розібрались, як працюють і отримують результат нейронні мережі для задачі апроксимації функцій, було розглянуто теоретичні основи апроксимації функцій, зокрема багатьох змінних, нейронними мережами прямого зв'язку, розширення теореми Стоуна, в тому числі універсальну апроксимаційну теорему, симплекційну апроксимаційну теорему, їх конструктивні доведення та передбачення явного алгоритму побудови мереж, а також їх використання для обґрунтування можливостей наближення нейронними мережами; питання узагальнення і максимального розширення класу задач, розв'язних нейронними мережами за рахунок використання триангульованих просторів, досліджено використання багат шарового перцептрона та його недоліків для задачі апроксимації, розглянуто і протестовано вдосконалену модель нейронної мережі, що використовує радіальні базисні функції у якості активаційних, і дозволяє апроксимувати доволі складну неперервну функцію на відрізку без значних затрат часу.

Література

- [1] Zarita Zainuddin, Ong Pauline. Function Approximation Using Artificial Neural Networks. International journal of systems applications, engineering development, Vol. 1, 2007.
- [2] Rocio Gonzalez-Diaz, Miguel A. Gutiérrez-Naranjo, Eduardo Paluzo-Hidalgo. Two-hidden-layer feedforward networks are universal approximators: a constructive approach. arXiv preprint arXiv:1907.11457, 2019.
- [3] Kailiang Wu, Dongbin Xiu. An explicit neural network construction for piecewise constant function approximation. arXiv preprint arXiv:1808.07390, 2018.
- [4] Yue Wu, Hui Wang, Biaobiao Zhang, K.-L. Du. Using Radial Basis Function Networks for Function Approximation and Classification. International Scholarly Research Network, ISRN Applied Mathematics, Vol. 2012, 2012.
- [5] J. Park and I. W. Sandberg. Universal Approximation Using Radial-Basis-Function Networks. Neural Computation, vol. 3, no. 2, pp. 246-257, 1991.
- [6] Горбань А. Н. Обобщенная аппроксимационная теорема и вычислительные возможности нейронных сетей. Сибирский журнал вычислительной математики, 1:1 (1998), с. 12–24.
- [7] Cybenko. G. Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals, and Systems, Vol. 2, 1989, PP. 303 - 314.

Додатки

Додаток А

```
import math
import matplotlib.pyplot as plt
import numpy as np

X = 0.00

def fnn(X):
    input = []
    k = 0.0000000000000001
    while k <= 1:
        input.append(k)
        k += 0.02
    n = len(input)

    hidden_l1 = [[0]*n for i in range(n)]
    for i in range(n):
        for j in range(n):
            hidden_l1[i][j] = X*(input[i] - input[j])
            - 0.5*(input[i] - input[j]) * (input[i] + input[j])
            if (hidden_l1[i][j] <= 0):
                hidden_l1[i][j] = 0
            else:
                hidden_l1[i][j] = 1
```

```

hidden_l2 = [0]*n
for i in range(n):
    for j in range(n):
        hidden_l2[i] += hidden_l1[i][j]
    hidden_l2[i] += 1-n
    if (hidden_l2[i] < 0):
        hidden_l2[i] = 0
    else:
        hidden_l2[i] = 1

appr_func = [0]*n
for i in range(n):
    appr_func[i] = math.sin(5*math.pi*input[i])
answer = 0
for i in range(n):
    answer += appr_func[i]*hidden_l2[i]
return answer

```

```

ask = []
t = 0.00000000000000000001
while t <= 1:
    ask.append(t)
    t += 0.01
res = [0.0]*len(ask)
for i in range(len(ask)):
    res[i] = fnn(ask[i])

```

```

xs = []
sin_vals = []
x = 0.0

```

```
while x < 1.0:
    sin_vals += [math.sin(5*math.pi*x)]
    xs += [x]
    x += 0.003

plt.scatter(xs, sin_vals, c='green', s=6)
plt.plot(xs, sin_vals, color = 'grey', linestyle = 'dashed')
plt.show()
```

Додаток Б

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import math
import random
from keras.models import Sequential
from tensorflow.keras.layers import Dense

x_train = []
for i in range(0, 10000):
    x = random.uniform(0, 10)
    x_train.append(x)

x_test = []
for i in range(0, 500):
    x = random.uniform(0, 10)
    x_test.append(x)

y_train = []
for x in x_train:
    y = math.sin(x)
    y_train.append(y)

plt.figure(figsize=(8,8))
plt.scatter(x_train, y_train, c = 'green', s = 5)

def make_model():
    model0 = Sequential()
```

```

model0.add(Dense(500, input_dim=1, activation='relu'))
model0.add(Dense(100, activation='relu'))
model0.add(Dense(1))

return model0

model0 = make_model()
model0.summary()

model0.compile(
    loss='mean_squared_error',
    optimizer='adam',
    metrics=['mse']
)

model0.fit(x_train, y_train, epochs=20, batch_size=10)

predictions = model0.predict(x_test)

plt.scatter(x_test, predictions, c = 'green', s = 4)

plt.figure(figsize=(8,8))
plt.scatter(x_train, y_train, c = 'grey', s = 5)
plt.scatter(x_test, predictions, c = 'green', s = 50,
marker = '.')
plt.show()

y_test_actual = []
for x in x_test:

```

```
y = math.sin(x)
y_test_actual.append(y)

from sklearn.metrics import mean_squared_error

mse = mean_squared_error(y_test_actual, predictions)
```

Додаток В

```
import numpy as np
from scipy.linalg import norm, pinv
from scipy import *
from matplotlib import pyplot as plt
import math
from sklearn.metrics import mean_squared_error

class RBFN:
    def __init__(self, indim, n_cent, outdim):
        self.indim = indim
        self.outdim = outdim
        self.n_cent = n_cent
        self.centers = [random.uniform(-1, 1, indim)
                        for i in range(n_cent)]
        self.b = 8
        self.W = random.random((self.n_cent, self.outdim))

    def basisfunc(self, c, d):
        assert len(d) == self.indim
        return exp(-self.b * norm(c-d)**2)

    def activation(self, X):
        G = zeros((X.shape[0], self.n_cent), float)
        for ci, c in enumerate(self.centers):
            for xi, x in enumerate(X):
                G[xi,ci] = self.basisfunc(c, x)
        return G
```

```

def train(self, X, Y):
    rand_idx = random.permutation(X.shape[0])
    [:self.n_cent]
    self.centers = [X[i,:] for i in rand_idx]
    G = self.activation(X)
    self.W = dot(pinv(G), Y)

def test(self, X):
    G = self.activation(X)
    Y = dot(G, self.W)
    return Y

n = 10000
x = []
for i in range(0, n):
    xx = random.uniform(0, 10)
    x.append(xx)
x = np.array(x)
x = x.reshape(n, 1)

x_test = []
for i in range(0, 500):
    xx = random.uniform(0, 10)
    x_test.append(xx)
x_test = np.array(x_test)
x_test = x_test.reshape(500, 1)

y = []
for xx in x:
    yy = math.sin(xx**2)*(math.cos(xx+2))**2

```

```
+(math.sin(xx))**3
y.append(yy)

rbfn = RBFN(1, 100, 1)
rbfn.train(x, y)
y_test = rbfn.test(x_test)

plt.figure(figsize=(15, 6))
plt.scatter(x, y, c='grey', s = 5)
plt.scatter(x_test, y_test, c='green', s = 20)
plt.show()
```

Додаток Г

```
n = 10000
xy = []
for i in range(0, n):
    x1 = random.uniform(0, 10)
    y1 = random.uniform(0, 10)
    pair = [x1, y1]
    xy.append(pair)
xy=np.array(xy)
xy.reshape(n, 2)

x_test = []
for i in range(0, 500):
    x1 = random.uniform(0, 10)
    y1 = random.uniform(0, 10)
    pair = [x1, y1]
    x_test.append(pair)
x_test=np.array(x_test)
x_test.reshape(500, 2)

y = []
for xx in xy:
    yy = math.sin(xx[0]+xx[1])
    y.append(yy)

rbfn = RBFN(2, 1000, 1)
rbfn.train(xy, y)
y_test = rbfn.test(x_test)
```

```

y_test_actual = sin(x_test)
y_test_actual = []
for xx in x_test:
    yy = math.sin(xx[0]+xx[1])
    y_test_actual.append(yy)

mse = mean_squared_error(y_test_actual, y_test)

from mpl_toolkits.mplot3d import axes3d
from matplotlib import cm

XP=[]
YP=[]
for t in x_test:
    XP.append(t[0])
    YP.append(t[1])
XP = np.array(XP)
YP = np.array(YP)

fig = plt.figure(figsize=(12, 10))
ax = fig.add_subplot(111, projection="3d")
X, Y = np.mgrid[0:10:200j, 0:10:200j]
Z = np.sin(X+Y)
ax.plot_surface(X, Y, Z, cmap=cm.YlGn, lw=0.5,
rstride=1, cstride=1)

ax.scatter(XP, YP, y_test, c='green', s = 40)
plt.show()

```