Mathematics in Modern Technical University
ISSN 2664-4258

Mathematics in Modern Technical University, 2025(2), 83 – 101

10.20535/mmtu-2025.2-083

Узагальнені лінійні моделі та нейронні мережі у страхуванні

А.А. Новікова[1], О.І. Василик[2]

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», м. Київ, Україна
[2]ORCID ID: https://orcid.org/0000-0002-0880-3751
(National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine)

Анотація

Основою забезпечення платоспроможності страхової компанії є актуарні розрахунки. В даній роботі розглядається нещодавно запропонована комбінована актуарна нейронна мережа, яка поєднує традиційну узагальнену лінійну модель, що використовується у страховому ціноутворенні, з нейронною мережею. Основна ідея використання нейронної мережі для ціноутворення у страхуванні полягає у моделюванні взаємодії між функціями, які не охоплюються узагальненою лінійною моделлю.

Ключові слова: страхування, ціноутворення, узагальнена лінійна модель, нейронна мережа, комбінована актуарна нейронна мережа, архітектура, вбудовування шарів, страхування автомобілів, модель регресії Пуассона.

MSC2010 62-07, 62J12
УДК 519.2, 368.91

a.novikova.meta@gmail.com, vasylyk@matan.kpi.ua

Cite as: Novikova, A.A. & Vasylyk, O.I. (2025). Generalized linear models and neural networks in insurance. Mathematics in Modern Technical University, 2025(2), 83-101. 10.20535/mmtu-2025.2-083

1 Вступ

Страхування базується на принципі, що група людей робить внески до спільного фонду для відшкодування втрат тих, хто постраждав від страхової події. На конкурентному ринку страховики можуть бути прибутковими лише тоді, коли їхні ціни якомога точніше відображають ризики, які вони покривають.

Незважаючи на те, що узагальнена лінійна модель (generalized linear model, GLM) є стандартним інструментом ціноутворення у страхуванні, крім страхування життя, починаючи з 1990-х років [6] дослідники та практики постійно прагнуть покращити продуктивність та ефективність процесу моделювання, нещодавно звернувшись до машинного навчання (machine learning, ML). Застосування методів машинного навчання в актуарній науці вивчалося і теоретично обґрунтовувалося ([4], [5]), зокрема і застосування нейронних мереж (neural networks, NN).

Основними перевагами моделі NN у визначенні страхових тарифів є чудова статистична продуктивність та автоматичне моделювання складних взаємодій між елементами [3]. Перша перевага ще не визначена науково, оскільки в літературі немає єдиної думки щодо неї ([3], [5], [7]), що вказує на те, що результати залежать від ситуації та застосування. Інша перевага є менш неоднозначною при моделюванні за допомогою GLM, оскільки моделювання взаємодій вручну є виснажливим процесом з обмеженою здатністю досліджувати складні взаємозв'язки. Таким чином, нейронні мережі потенційно забезпечують подвійне покращення порівняно з GLM у цій галузі з точки зору часу та продуктивності.

Незважаючи на вищезазначені переваги, використання моделей ML і NN ще не набуло широкого розповсюдження в комерційному страхуванні, головним чином через їхній основний недолік – відсутність інтерпретації. Моделі NN часто називають моделями "чорної скриньки", оскільки їх можна аналізувати лише на вхідних-вихідних даних, а не на внутрішній роботі. Зокрема, у той час як GLM призначає ваги функціям, вказуючи таким чином на вплив функції на прогнози в простій (зазвичай мультиплікативній) моделі, моделі ML просто створюють прогноз, не пояснюючи, як

модель прийшла до такого висновку. Страхові компанії не тільки хочуть розуміти, як їхній ризик розподіляється між клієнтами, але також повинні мати можливість пояснити, що вплинуло на кінцевий розмір страхової премії клієнта. Існують також проблеми впровадження від аналітичного робочого процесу до виробничої системи за відсутності чітко визначеної формули ціноутворення. Ще однією проблемою NN моделей вважається неоднозначність ціноутворення найкращих моделей ML за певного сценарію. На відміну від GLM, дві моделі NN можуть, враховуючи стохастичність процесу навчання, мати однакову статистичну продуктивність, але давати різні індивідуальні прогнози для того самого набору даних [2].

Визначивши переваги та недоліки нейронних мереж, була запропонована нова модель, яка поєднує класичну GLM із нейронною мережею, в якій намагались зберегти переваги обох [8]. Таку модель назвали комбінованою актуарною нейронною мережею (CANN). На практиці передбачається, що модель GLM може бути вкладена в NN за допомогою пропуску з'єднання, щоб створити вдосконалення моделі GLM, не відхиляючись надто далеко від свого оригіналу. Це також дозволяє NN досліджувати взаємодію між функціями, які не розглядалися в GLM. Було запропоновано кілька конфігурацій моделі CANN, де розробник моделі може вирішити, чи повинен параметр моделі GLM бути навчальним чи ні в NN [8].

Підхід CANN дозволив покращити статистичні показники GLM-моделей, оскільки відсутні взаємодії можна систематично ідентифікувати [1]. Крім того, є деякі інші переваги запропонованої моделі, а саме автоматизований процес вибору параметрів і швидка збіжність алгоритму градієнтного спуску, що дозволяє використовувати методи бутстрепу (оцінювання на основі інших оцінок) для вивчення точності прогнозу [10].

Підхід CANN зберігає перевагу нейронної мережі в автоматичному визначенні взаємодії складних функцій, а також зменшує «неоднозначність найкращої моделі», оскільки він походить від моделі GLM. Однак основний недолік моделі CANN все ще є: відсутність інтерпретації для NN зберігається в моделі CANN. Таким чином, ключовим питанням для оцінки життєздатності моделей «чорної скриньки» для

страхових спеціалістів є краще розуміння того, як їх можна розшифрувати в актуарному контексті. Вплив окремих функцій і взаємодія функцій – це дві ключові сфери, які потребують певного пояснення. В науковій літературі представлено численні методи та інструменти, які спрямовані на інтерпретацію моделей «чорної скриньки» ([11], [12]). Оцінка їх релевантності в актуарних умовах для моделей CANN є важливою частиною оцінки корисності моделей CANN.

В роботі [13] автори пропонують свій погляд на те, як останніми роками великий потік даних у поєднанні з постійним розвитком інформаційних технологій і народженням науки про дані революціонізував більшість сфер актуарної науки, а також практики страхування. Зокрема, вони показують, як у випадку загального страхування комбінація класичних інструментів статистики, таких як узагальнені лінійні моделі, та новітніх інструментів, таких як, наприклад, нейронні мережі, сприяє кращому розумінню та аналізу актуарних даних.

Дана стаття підготовлена за матеріалами магістерської дисертації Новікової А.А. [21], в якій досліджуються узагальнені лінійні моделі та нейронні мережі з точки зору їх застосування у страхуванні, побудовано деякі узагальнені лінійні моделі, нейронні мережі з вбудованими компонентами та комбіновані актуарні нейронні мережі в середовищі RStudio, виконано порівняльний аналіз цих моделей. Дисертація [21] містить відповідні скрипти та результати обчислень.

2 Узагальнені лінійні моделі

2.1. Визначення

Узагальнена лінійна модель (generalized linear model, GLM) — це клас регресійних моделей, визначених загальним чином, за допомогою яких можна змоделювати зв’язок між залежною змінною y_i та d незалежними змінними $x_i \in R^d$ з використанням невідомого параметра $\beta \in R^d$. Ключовим припущенням GLM є те, що розподіл залежної змінної є членом сімейства експоненційних розподілів.

Прикладами є розподіл Гауса, біноміальний, Пуассона, експоненційний і гамма розподіл. Загальний вигляд цих розподілів такий

$$f(y_i, \theta_i, \varphi) = exp\left(\frac{y_i\theta_i - b(\theta_i)}{a(\varphi)} + h(y_i, \varphi)\right) \quad (2.1)$$

де φ — параметр масштабу, а θ_i — параметр розташування. Наступні ключові властивості мають місце для членів експоненційного сімейства,

$$\mu = E[y] = \frac{db(\theta_i)}{d\theta_i}, \quad (2.2)$$

$$Var(y) = \frac{d^2b(\theta_i)}{d\theta_i^2} = \frac{d\mu}{d\theta_i}a(\varphi). \quad (2.3)$$

Основною ідеєю GLM є лінійна модель відповідної функції математичного сподівання залежної змінної. Зазвичай це визначається лінійним предиктором η_i,

$$\eta_i = g(E[y_i]) = g(\mu_i) = \langle x_i, \beta \rangle \quad (2.4)$$

Очікувана відповідь або прогноз, таким чином, визначається так:

$$E[y_i] = g^{-1}(\eta_i) = g^{-1}(\langle x_i, \beta \rangle) \quad (2.5)$$

Функція g називається функцією зв'язку, оскільки вона пов'язує лінійний предиктор з очікуваною відповіддю. Таким чином, GLM визначається двома варіантами моделювання: вибором розподілу залежної змінної та вибором функції зв'язку. Зазвичай функція зв’язку вибирається так, щоб $\eta_i = \theta_i$ для обраної функції розподілу. Методом оцінювання невідомого вектора β є оцінка максимальної правдоподібності (MLE) з алгоритмом, який ґрунтується на ітераційному повторно зваженому методі найменших квадратів [14].

2.2. Узагальнені лінійні моделі в актуарній математиці

Існує дві основні причини використання GLM для визначення страхових тарифів. По-перше, GLM працюють із рядом розподілів із сімейства експоненційних, таких як розподіл Пуассона та гамма-розподіл. По-друге, модель для середнього є не

просто лінійною функцією незалежних змінних, як у випадку простої лінійної регресії, а скоріше монотонним перетворенням середнього. Певні розподіли частіше використовуються при визначенні страхових тарифів. Розподіл Пуассона доцільно використовувати для моделювання кількості позовів, що надходять до страхової компанії. Розподіл Твіді рекомендується для моделювання чистої премії, а гамма-розподіл рекомендовано для налаштування «серйозності» позову тощо [6].

2.3. Від узагальнених лінійних моделей до нейронних мереж

Більшість методів моделювання в актуарному ціноутворенні, які використовуються сьогодні, базуються на фундаментальних роботах [15], [16] про узагальнені лінійні моделі (GLM). Наразі актуарне ціноутворення в автострахуванні зазвичай базується на 40–50 коваріатах, які розрізняють страхувальників. За останні кілька десятиліть актуарії набули великого практичного досвіду в розробці інформації, яка може бути корисною для прогнозного моделювання в GLM. Актуаріям доводиться мати справу з кількома основними проблемами. По-перше, більшість пояснювальних змінних є категоріальними, і, як наслідок, статистичний аналіз стикається зі складними проблемами, наприклад, розрідженістю базової розрахункової матриці. Крім того, в регресійних функціях коваріати взаємодіють у нетривіальний спосіб, що робить належну оцінку складним завданням. Моделювання частоти страхових випадків — це проблема прогнозування рідкісних подій (проблема дисбалансу класів), де актуарії намагаються знайти систематичні ефекти в даних, у яких значною мірою переважає шум (випадкова частина). Оскільки не існує простої готової моделі розподілу, моделювання розміру страхових виплат має на меті знайти хороший компроміс між складністю та точністю моделі. Це справедливо, наприклад, в сімействі розподілів експоненціальної дисперсії (EDF), які зазвичай не відповідають всьому діапазону розмірів вимог. Це призвело до того, що широко досліджуються все більш складні моделі, що призводить до технічних ускладнень (див., наприклад, [17], [18], [19]).

3. Нейронні мережі

3.1. Ідея нейронних мереж

Для вирішення завдань обробки даних все частіше використовуються методи машинного навчання. Постійно зростаючі бази даних ускладнюють «ручну» розробку моделей, змушуючи актуаріїв все більше покладатися на вивчення таких інструментів, як нейронні мережі. Існують активні спільноти актуаріїв, які стимулюють дослідження в цій галузі, наприклад, ініціатива з науки про актуарні дані Швейцарського інституту актуаріїв.

Нейронна мережа (neural networks, NN) прямого поширення — це модель статистичного навчання, назва якої походить від нейронної структури мозку, яку вона імітує. Метою NN прямого поширення є апроксимація деякої функції $y = f(x, \theta)$, де вхідний сигнал x використовується для прогнозування вихідного сигналу y. При оцінці параметрів розглядається параметр мережі θ, який дає найкраще наближення функції. Термін «пряме поширення» означає, що інформація передається тільки вперед у мережі, без зворотного зв'язку на виході з будь-якого рівня. В основі NN лежить теорема про універсальне наближення, яка стверджує, що NN із принаймні одним прихованим шаром, в якому задано достатньо прихованих одиниць, може апроксимувати будь-яку неперервну функцію з певними обмеженнями [20].

3.2. Структура нейронної мережі та визначення

Термін «мережа» стосується способу моделювання функції f шляхом об'єднання функцій. Наприклад, у мережі, що складається з двох шарів, перший шар можна представити як $f^{(1)}$ і другий як $f^{(2)}$. Тоді складена форма цих шарів буде $f(x) = f^{(2)}\left(f^{(1)}(x)\right)$. У NN існує три типи шарів: вхідний шар, приховані шари та вихідний шар. У наведеному вище прикладі x буде вхідним шаром, $f^{(1)}$ буде прихованим шаром, а $f^{(2)}$ буде вихідним шаром. Таким чином, приховані шари є проміжними між вхідним і вихідним шарами. Кожен шар є вектором із заданою

розмірністю, який отримує вхідні дані від попереднього шару, а потім відображає ці значення на наступному шарі. Опис структури NN зазвичай передбачає визначення глибини та ширини моделі; в нашому випадку глибина — це кількість шарів, а ширина вказана для кожного шару як розмірність цього шару.

Вибір глибини та ширини мережі є складною задачею. Згідно з теоремою про універсальне наближення, для представлення будь-якої функції достатньо одного шару. Однак ширина такого шару може бути дуже великою, і, відповідно, модель буде складною для навчання. Загалом більш глибокі мережі дають можливість зменшити ширину, необхідну для наближення функції [20]. Знаходження оптимального поєднання глибини та ширини мережі зазвичай досягається шляхом спроб і помилок.

Функції, які відображають кожен шар на інший, називаються прихованими одиницями, а функція активації, яка відображає останній прихований шар на вихідний, називається функцією виведення. Їх може вибрати розробник моделі, і для цього існує декілька варіантів. Враховуючи вхідні дані x, пристрій зазвичай спочатку обчислює афінне перетворення $z = \omega^T x + b$, а потім застосовує поелементну нелінійну функцію $\varphi(z)$. Тут ω, b називаються вагами та зміщеннями і, таким чином, представляють параметри мережі. Крім того, $\varphi(z)$ називається функцією активації, яка зазвичай відрізняє тип прихованої одиниці. Одиниця для вихідного рівня називається вихідною одиницею, і її потрібно вибрати відповідно до типу та розподілу залежної змінної, здебільшого залежно від того, чи є завдання класифікації чи регресії. Трьома найпоширенішими варіантами для прихованих одиниць є випрямлені лінійні одиниці, логістична сигмоїдна одиниця та гіперболічний тангенс. Важливою особливістю цих одиниць є градієнт, оскільки він використовується для калібрування параметрів мережі [20].

Випрямлені лінійні одиниці (Rectified Linear Units, ReLU) використовують функцію активації $\varphi(z) = max\{0, z\}$. Градієнт ReLU залишається великим, коли блок активний, що сприяє процедурі навчання. Більше того, це дозволяє розріджену

активацію, оскільки одиницю можна легко встановити на нуль. Це означає, що мережа, як правило, менш затратна з точки зору обчислень, ніж мережа, яка використовує інші приховані одиниці. Недоліком методу ReLU є те, що він не може навчатися із спостережень, для яких їх активація дорівнює нулю. Загальним підходом для усунення цього недоліку є використання ReLU з витоком, який фіксує константу α до малого значення, наприклад $\alpha = 0.001$, і визначає функцію активації як $\varphi(z) = max\{\alpha z, z\}$.

Сигмоїдна одиниця використовує сигмоїдну функцію як функцію активації, $\varphi(z) = \sigma(z) = \frac{1}{1+e^{-x}}$, а гіперболічний тангенс використовує функцію гіперболічного тангенса $\varphi(z) = th(z)$. Ці дві функції пов'язані $th(z) = 2\sigma(2z) - 1$. Сигмоїдна функція має діапазон $(0,1)$, а функція гіперболічного тангенса – $(-1,1)$. Градієнт цих функцій завжди додатний, близький до лінійного поблизу нуля, але асимптотично спадає. Таким чином, ці одиниці дозволяють прихованим одиницям по суті стати класифікаторами, придатними для певних проблем класифікації. Проте асимптотично спадаючі градієнти створюють проблему при навчанні моделей, оскільки одиниці часто насичуються до низького значення, коли z від'ємне, і до високого значення, коли z додатне [20].

Враховуючи наведені вище визначення, дамо формальне визначення NN прямого поширення. Нехай x_i — вектор вибірок k незалежних змінних для спостереження i, а y_i — вибірка залежної змінної. Розглянемо NN з $l = 1, \dots, L$ прихованих шарів, де шар lскладається з $q^{(l)}$ прихованих одиниць, позначених $a_q^{(l)}$, де $q = 1, \dots, q^{(l)}$ і $a^{(l)} = \left\{a_1^{(l)}, \dots, a_{q^{(l)}}^{(l)}\right\}$. Припустимо, що мережа використовує функцію активації $\varphi(z)$ і афінне перетворення

$$z_q^{(l)} = \left\langle \omega_q^{(l)}, a^{(l-1)} \right\rangle + b_q^{(l)} \tag{3.1}$$

таке, що

$$a_q^{(l)} = \varphi\left(z_q^{(l)}\right) = \varphi\left(\left\langle \omega_q^{(l)}, a^{(l-1)} \right\rangle + b_q^{(l)}\right), \tag{3.2}$$

для $q = 1, \dots, q^{(l)}$, де $\omega_q^{(l)}$ це $q^{(l-1)}$-вимірний вектор ваг від прихованих одиниць у $(l-1)$-му шарі до q-го прихованого блоку в l-му шарі, а $b_q^{(l)}$ — зміщення для q-го прихованого блоку в l-му шарі. Це позначення можна додатково спростити, позначивши

$$W^{(l)} = \left\{\omega_1^{(l)}, \dots, \omega_{q^{(l)}}^{(l)}\right\} \in R^{q^{(l-1)} \times q^{(l)}} \tag{3.3}$$

$$b^{(l)} = \left\{b_1^{(l)}, \dots, b_{q^{(l)}}^{(l)}\right\} \tag{3.4}$$

і нехай $\varphi(z)$ є поелементною функцією, тому

$$a^{(l)} = \varphi\left(W^{(l)T} a^{(l-1)} + b^{(l)}\right) \tag{3.5}$$

Перший рівень прихованих одиниць, $a^{(l)}$ отримує x_i як вхідні дані, а всі наступні рівні $a^{(l)}$ отримують інформацію від $a^{(l-1)}$. Останній шар є просто вихідним рівнем, а також передбаченням $\hat{y}_i = a^{(L+1)}$. На рисунку 1 показано спрощену ілюстрацію NN з виділенням вхідного рівня, прихованих одиниць в одному прихованому шарі та вихідного рівня.

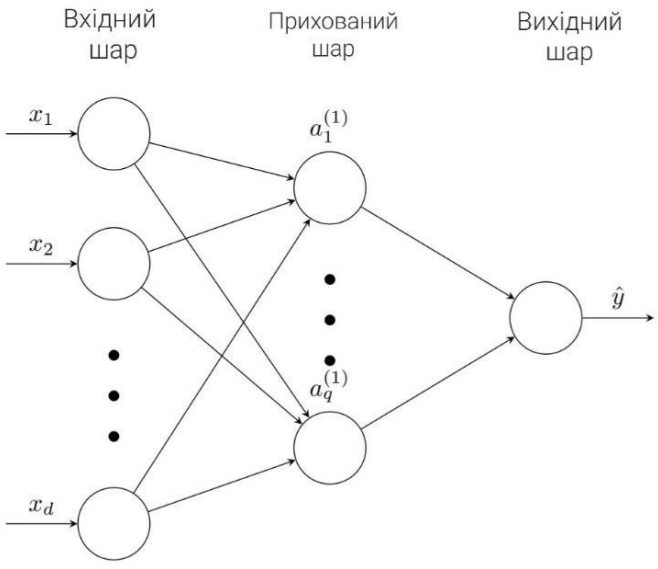

Рис.1: Ілюстрація NN з d входами, прихованим шаром з глибиною q та одним виходом $\hat{y}$.

3.3. Алгоритм навчання мережі

Як і більшість моделей машинного навчання, нейронні мережі використовують градієнтну оптимізацію функції втрат (loss function) для оцінки параметра мережі $\theta = \{W^{(1)}, \dots, W^{(L+1)}, b^{(1)}, \dots, b^{(L+1)}\}$. Тобто, градієнти функції втрат по відношенню до кожної ваги та зміщення використовуються для оновлення параметра мережі з метою мінімізації функції втрат. Нелінійність нейронних мереж призводить до того, що більшість функцій втрат стають неопуклими, тому алгоритми на основі градієнта лише зменшують функцію втрат до низького значення, але не обов'язково забезпечують збіжність до глобального мінімуму. Існує декілька функцій втрат, і їх придатність залежить від проблеми, що моделюється. Зазвичай в задачах регресії використовується середня квадратична похибка (MSE):

$$C(\hat{y}_i, y_i) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{3.6}$$

де n — кількість значень у вибірці, y_i — спостережуваний результат, а $\hat{y}_i$— прогнозований результат. Крім того, у випадках, коли припускається, що залежна змінна походить з даних обчислень, можна використовувати функцію втрат Пуассона [20]:

$$C(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i log\hat{y}_i). \tag{3.7}$$

Таким чином, у структурі мережі важливим завданням градієнтного навчання є обчислення всіх градієнтів ваг і зміщень для функції втрат. Це робиться за допомогою методу зворотного поширення, який обчислює градієнти мережі для однієї навчальної точки даних за допомогою ланцюгового правила числення. Для заданої точки вхідних даних спочатку обчислюються значення всіх прихованих одиниць у мережі шляхом поширення вперед у мережі. На вихідному рівні функція втрат обчислюється з використанням спостережуваного та прогнозованого результату

мережі. Потім обчислюються градієнти в кожній із прихованих одиниць, використовуючи правило ланцюгових обчислень, зворотного поширення через мережу. У контексті мережі, визначеної в пункті 3.2, член похибки для кожного прихованого блоку в мережі визначається як часткова похідна функції втрат відносно зваженого входу $z^{(l)}$:

$$\nabla_{z^{(l)}} C = \delta^{(l)}. \tag{3.8}$$

Використовуючи ланцюгове правило числення,

$$\delta^{(l)} = \nabla_{a^{(l)}} C \odot \varphi'\left(z^{(l)}\right). \tag{3.9}$$

Похибка допускає прості вирази шуканих градієнтів $\nabla_{W^{(l)}} C$ та $\nabla_{b^{(l)}} C$, оскільки за правилом ланцюга

$$\nabla_{b^{(l)}} C = \nabla_{z^{(l)}} C \odot \nabla_{b^{(l)}} z^{(l)} = \delta^{(l)} \tag{3.10}$$

та

$$\nabla_{W^{(l)}} \mathrm{C} = \nabla_{z^{(l)}} C \odot \nabla_{W^{(l)}} z^{(l)} = a^{(l-1)} \odot \delta^{(l)}, \tag{3.11}$$

де рівняння (3.8) було використано разом із визначенням $z^{(l)}$ і, таким чином, часткових похідних відносно $W^{(l)}$та $b^{(l)}$.

При навчанні мережі та оновленні ваг і зміщень для кожного рівня наближені похідні не застосовуються безпосередньо до попередніх значень, а масштабуються за допомогою параметра швидкості навчання. Параметр швидкості навчання зазвичай вибирається в діапазоні (0, 1], і масштабує довжину кроків, зроблених у напрямку наближення, мінімізуючи функцію втрат. Загалом, велика швидкість навчання зменшує час навчання, але ціною підвищеного ризику перевищення оптимуму. Зазвичай тестують набір параметрів швидкості навчання та аналізують відповідні результати, щоб знайти підходящу швидкість навчання для конкретної моделі та проблеми [20].

Оскільки навчання нейронних мереж часто використовує великі набори даних для якісного узагальнення, то обчислення градієнта для всього набору даних за один прогін є дорогим з обчислювальної точки зору. Стохастичний градієнтний спуск (Stochastic Gradient descend, SGD) використовується майже у всіх версіях нейронних мереж, оскільки він розбиває навчальний набір даних розміром n на m пакетів. Градієнт для кожного пакета обчислюється окремо та застосовується для оновлення параметра мережі θ для всіх m пакетів. З математичної точки зору, SGD полягає в тому, що градієнт — це математичне сподівання, яке оцінюється на основі даних з цих пакетів. Зазвичай розмір пакету коливається від одного до кількох сотень, незалежно від розміру всього набору даних. Загалом менші розміри пакетів корелюють із кращим узагальненням моделі, можливо, через шум, який вони додають до процесу навчання. Крім того, через високу дисперсію, що виникає при використанні малої кількості даних в пакеті, малі розміри пакетів зазвичай супроводжується невеликою швидкістю навчання для підтримки стабільності навчання. На практиці алгоритм навчання виконується кілька разів для всього набору даних, при цьому кожен запуск називається епохою. Таким чином, у кожну епоху набір даних сегментується на випадкові пакети, а потім подається в мережу за допомогою алгоритму SGD [20].

3.4. Регуляризація

Нейронні мережі, як і багато інших алгоритмів машинного навчання, схильні до перенавчання (англ. overfitting). Коли відбувається перенавчання, модель зазвичай працює виключно добре на навчальному наборі даних, але погано для будь-якого тестового набору даних, оскільки в помилці оцінювання домінує дисперсія, а не зміщення [20]. Методи регуляризації відносяться до будь-якого методу, який зменшує помилку тесту, можливо, в обмін на збільшену помилку навчання. Найпоширеніші з них включають регуляризацію L1, регуляризацію L2 та ранню зупинку [20].

3.5. Скіп-з'єднання

Скіп-з'єднання (skip-connection)— це будь-яке з'єднання, яке дозволяє інформації перетікати з рівня t на рівень l, де $t \neq l-1$, оскільки $t = l-1$ визначає звичайний спосіб поширення інформації через мережу. У математичних термінах $a^{(t)}$ надається як вхідні дані для $a^{(l)}$ на додаток до звичайного члена $z^{(l)} = W^{(l)}a^{(l-1)} + b^{(l)}$ через функцію $s\left(a^{(t)}\right)$, так що $a^{(l)} = \varphi\left(z^{(l)} + s\left(a^{(t)}\right)\right)$. Найпоширенішим скіп-з'єднанням, що використовується, є тотожне відображення

$$s\left(a^{(t)}\right) = \left\langle l^{q^{(l)}}, a^{(t)} \right\rangle = \sum_{q=1}^{q(t)} a_q^{(t)}. \tag{3.12}$$

Скіп-з'єднання вперше були запропоновані як метод моделювання нейронних мереж у статті про мережі глибокого навчання для розпізнавання зображень [9], які виявили, що такі з'єднання полегшують оптимізацію більш глибоких нейронних мереж, зберігаючи статистичне підвищення продуктивності від збільшення глибини.

4. Комбінована актуарна нейронна мережа (CANN)

4.1. Структура мережі

Розглянемо GLM із функцією зв'язку $g(E[y_i]) = g(\mu_i) = \langle x_i, \beta \rangle$, так що $\mu_i = g^{-1}(\langle x_i, \beta \rangle)$. Модель передбачає розподіл із експоненційного сімейства, властивості якого не важливі на даному етапі. Припустимо, враховуючи цю інформацію, що модель навчена і ефективну оцінку максимальної правдоподібності $\hat{\beta}^{MLE}$ знайдено. Прогноз $\hat{y}_i^{GLM}$ для заданого спостереження i визначається як $\hat{y}_i^{GLM} = g^{-1}(\langle x_i, \beta^{MLE} \rangle)$.

Розглянемо тепер NN прямого поширення з L прихованих одиниць, $l = 0,1,\dots,L,L+1$, де $a^{(l)}$ позначає l-й рівень прихованих одиниць і $a^{(0)} = x_i$, $W^{(l)}$ – матриця ваг $q^{(l-1)} \times q^{(l)}$, $b^{(l)}$ – вектор зсуву для шару l, а також введемо функції активації φ для прихованих шарів і ψ для вихідного шару. Крім того, нехай між вхідним шаром із ваговою матрицею $\hat{\beta}^{MLE}$ та вихідним шаром існує пропускний зв'язок, такий, що

$$a^{(L+1)} = \psi\left(W^{(L+1)}a^{(L)} + b^{(L+1)} + \left\langle x_i, \hat{\beta}^{MLE}\right\rangle\right) \tag{4.1}$$

Якщо параметр мережі ініціалізовано так, що всі ваги та зміщення дорівнюють 0, а ψ вибрано так, що $\psi(x) = g^{-1}(x)$, тоді

$$W^{(L+1)}a^{(L)} + b^{(L+1)} = 0 \tag{4.2}$$

та

$$a^{(L+1)} = \psi\left(\left\langle x_i, \hat{\beta}^{MLE}\right\rangle\right) = g^{-1}\left(\left\langle x_i, \hat{\beta}^{MLE}\right\rangle\right) = \hat{y}_i^{GLM}. \tag{4.3}$$

Таким чином, мережа ініціалізується для точного прогнозування GLM. Інтуїтивно зрозуміло, що будь-які коригування, які алгоритм навчання вносить до параметра $\theta = \left\{W^{(1)}, \dots, W^{(L+1)}, b^{(1)}, \dots, b^{(L+1)}\right\}$, будуть прагнути виправити помилки або залишки GLM.

Що стосується функції втрат, в [1] моделюють частоту претензій і їх GLM передбачає розподіл Пуассона і, отже, використовують пуассонівську функцію втрат. Немає детального обговорення щодо вибору функції втрат для мережі, але пуассонівська функція втрат також може бути використана для мережі.

4.2. Зв'язок між функцією зв'язку та функцією активації виходу

Ми описали вибір функції зв'язку та відповідної функції активації. У [1] та [8] обговорюють і використовують лише функцію лог-зв'язку $g(x) = log(x)$, і відповідно, функція активації виходу має вигляд $\psi(x) = exp(x)$. Ця конкретна функція активації гарним чином пов'язує GLM і NN, де частина GLM просто приписується фактору з NN. Нехай навчений параметр мережі позначено через $\hat{\theta}$ і для кожного $l = 1, \dots, L$ нехай $\hat{W}^{(L)}$і$\hat{b}^{(L)}$ будуть навченими вагами та зміщеннями для мережі відповідно. Крім того, нехай $\hat{y}_i^{NN} = \psi\left(\hat{W}^{(L)}a^{(L-1)} + b^{(L)}\right)$ позначає прогноз NN, а $\hat{y}_i^{CANN}$— прогноз CANN. З експоненціальною функцією активації виходу маємо

$$\hat{y}_i^{CANN} = exp\left(\hat{W}^{(L+1)}a^{(L)} + b^{(L+1)} + \left\langle x_i, \hat{\beta}^{MLE}\right\rangle\right) =$$

$$exp\left(\hat{W}^{(L+1)}a^{(L)} + b^{(L+1)}\right)exp\left(\left\langle x_i, \hat{\beta}^{MLE}\right\rangle\right) = \hat{y}_i^{NN}\hat{y}_i^{GLM}. \qquad (4.4)$$

Результат у рівнянні (4.4) показує, що при використанні функції лог-зв'язку для GLM і експоненціальної функції активації виходу для частини мережі прогноз моделі CANN можна розділити на два множники: коефіцієнт для частини GLM і коефіцієнт для частини NN. Отже, якщо припустити, що GLM робить точні прогнози, то коефіцієнт $\hat{y}_i^{NN}$ буде близьким до 1.

4.3. Вимірювання похибок

У цьому розділі розглядаються три методи вимірювання похибок для оцінювання продуктивності моделі: середня квадратична похибка (MSE), середня абсолютна похибка (MAE) і функція втрат Пуассона. Розглянемо набір даних з n точками даних і з емпіричними чистими преміями $y \in R^n$, для яких модель зробила прогнози $\hat{y} \in R^n$. Середня квадратична похибка визначається наступним чином:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2. \qquad (4.5)$$

MSE розглядає всі точки даних однаково, що може бути проблематично в ситуаціях, коли деякі точки даних становлять більший інтерес, ніж інші. Щоб переконатися, що квадратичні похибки були зважені відповідно до їх пропорційної ваги в наборі даних, ми визначаємо зважену середню квадратичну похибку (WMSE) як:

$$WMSE = \frac{1}{\sum_i^n v_i}\sum_{i=1}^{n} v_i(y_i - \hat{y}_i)^2, \qquad (4.6)$$

де v_i — експозиція або вага для точки даних i. MSE є загальним вибором для вимірювання похибки в задачах регресії, оскільки вона зводить похибки в квадрат, що завжди дає додатну похибку. Недоліком MSE є те, що вона дуже чутлива до викидів. Оцінка похибки, яка є більш стійкою до викидів, це середня абсолютна похибка (MAE), яка визначається як:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|. \quad (4.7)$$

Аналогічно до наведеного вище визначення WMSE, зважена середня абсолютна похибка (WMAE) визначається наступним чином:

$$WMAE = \frac{1}{\sum_i^n v_i}\sum_{i=1}^{n} v_i|y_i - \hat{y}_i|. \quad (4.8)$$

Пуассонівська функція втрат визначається як:

$$Loss = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i log\hat{y}_i), \quad (4.9)$$

а також зважені втрати, або WLoss визначаються наступним чином:

$$WLoss = \frac{1}{\sum_i^n v_i}\sum_{i=1}^{n} v_i(\hat{y}_i - y_i log\hat{y}_i). \quad (4.10)$$

Пуассонівська функція втрат часто використовується при моделюванні розподілу Пуассона, наприклад, в узагальнених лінійних моделях, а також при моделюванні нейронних мереж.

Висновки

У цій статті розглядаються узагальнені лінійні моделі та нейронні мережі з точки зору їх застосування у страхуванні. А саме, наведено підхід до вдосконалення класичних моделей регресії, які використовуються в актуарних розрахунках, за допомогою нейронних мереж. Розглядається модель комбінованої актуарної нейронної мережі, яка передбачає вкладення класичної параметричної регресійної моделі в архітектуру нейронної мережі для використання обох методів одночасно.

В результаті дослідження встановлено, що з точки зору похибки прогнозу використання комбінованої актуарної нейронної мережі призводить до покращення моделі в порівнянні з узагальненими лінійними моделями та нейронними мережами.

References:

1. J. Schelldorfer, M.V. Wüthrich. Nesting classical actuarial models into neural networks. 2019.
2. A. Gustafsson, J. Hansen. Combined Actuarial Neural Networks in Actuarial Rate Making. https://kth.diva-portal.org/smash/get/diva2:1656340/FULLTEXT01.pdf
3. C. Dugas, Y. Bengio, N. Chapados, P. Vincent, G. Denoncourt, and C. Fournier. Statistical learning algorithms applied to automobile insurance ratemaking. 12 2003. doi: 10.1142/9789812794246_0004.
4. R. Richman. AI in actuarial science – a review of recent advances – part 1. Annals of Actuarial Science, page 1–23, 2020. doi: 10.1017/S1748499520000238.
5. C. Blier-Wong, H. Cossette, L. Lamontagne, and E. Marceau. Machine learning in P&C insurance: A review for pricing and reserving. Risks, 9(1), 2021. doi: 10.3390/risks9010004.
6. E. Ohlsson, B. Johansson. Non-Life Insurance Pricing with Generalized Linear Models. EAA Lecture Notes. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
7. C. Mano, E. Rasa. A discussion of modeling techniques for personal lines pricing". Trans 27th ICA, 2002.
8. M.V. Wüthrich, M. Merz. Generalized linear models. ASTIN Bulletin: The Journal of the IAA, 49:1–3, 2019.
9. K. He, X. Zhang, S. Ren, J. Sun. Deep residual learning for image recognition, 2015.
10. A. Gabrielli, R. Richman, and M.V. Wüthrich. Neural network embedding of the over-dispersed poisson reserving model. Scandinavian Actuarial Journal, 2020(1):1–29, 2020. doi: 10.1080/03461238.2019.1633394.
11. R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. A survey of methods for explaining black box models. ACM Comput. Surv., 51(5), August 2018. doi: 10.1145/3236009
12. C. Molnar. Interpretable Machine Learning. Creative Commons License, 2020.
13. P. Embrechts and M.V. Wüthrich. Recent Challenges in Actuarial Science. https://docslib.org/doc/353687/recent-challenges-in-actuarial-science

14. D.C. Montgomery, E. A. Peck, and G. G. Vining. Introduction to linear regression analysis, volume 821 of Wiley series in probability and statistics. Wiley, Hoboken, 5th ed edition, 2012.
15. Nelder J.A, Wedderburn R.W.M. 1972. Generalized linear models. J. R. Stat. Soc. Ser. A 135: 370–84.
16. McCullagh P., Nelder J.A. 1983. Generalized Linear Models. London: Chapman & Hall.
17. Lee SCK, Lin XS. 2010. Modeling and evaluating insurance losses via mixtures of Erlang distributions. N. Am. Actuar. J. 14:107–30.
18. Miljkovic T., Grün B. 2016. Modeling loss data using mixtures of distributions. Insur. Math. Econ. 70:387–96.
19. Yin C, Lin XS. 2016. Efficient estimation of Erlang mixtures using iSCAD penalty with insurance application. ASTIN Bull. 46:779–99.
20. Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning. MIT Press, 2016. http: //www.deeplearningbook.org.
21. Novikova, A. A. Application of neural networks in actuarial calculations: master's thesis: 111 "Mathematics" / Novikova Alla Anatoliivna. – Kyiv, 2023. – 50 p. (Scientific supervisor: Vasylyk O.I.)

A.A. Novikova, & O.I. Vasylyk (2025). Generalized linear models and neural networks in insurance. *Mathematics in Modern Technical University*, 2025(2), 83-101.

Submitted: 2023-05-25
Accepted: 2023-06-30

Abstract. Actuarial calculations form the basis for ensuring the solvency of the insurance company. This paper is devoted to the study of recently proposed combined actuarial neural network that combines the traditional generalized linear model used in insurance pricing with a neural network. The main idea behind using a neural network for insurance pricing is to model interactions between functions that are not covered by a generalized linear model.

Key words: insurance, pricing, generalized linear model, neural network, combined actuarial neural network, architecture, embedding layers, car insurance, Poisson regression model.