

АНАЛІЗ МЕТОДІВ ПОЯСНОСТІ НЕЙРОННИХ МЕРЕЖ ДЛЯ ПОДОЛАННЯ ПРОБЛЕМИ ЧОРНОГО ЯЩИКА

Вербицький О.С.¹

¹Національний університет кораблебудування імені адмірала Макарова

¹Україна, Миколаїв

¹soomoalex@gmail.com

Анотація. *Рішення проблеми "чорного ящика" в штучному інтелекті критично важливе для забезпечення надійності, передбачуваності та розуміння роботи системи. Цей напрямок відіграє ключову роль у встановленні довіри до ШІ в медицині, правосудді та інших відповідальних сферах. У даній роботі досліджуються методи пояснюваності та прозорості нейронних мереж та їх вплив на створення надійних ШІ. Ключові слова: нейронні мережі; штучний інтелект; проблема чорного ящика; методи інтерпретації; пояснюваність систем.*

Вступна частина. Сучасні нейронні мережі є основою багатьох систем штучного інтелекту (ШІ), що використовуються в різних галузях — від медицини до фінансового аналізу. Однак одна з головних проблем таких моделей полягає у їхній непрозорості: внутрішні процеси прийняття рішень часто залишаються невідомими навіть розробникам, що створює явище так званого «чорного ящика». Це може ускладнити довіру до технологій і зробити їх результати непередбачуваними, особливо в критично важливих сферах, таких як охорона здоров'я чи правосуддя. Для подолання цієї проблеми необхідно розробляти й застосовувати методи пояснюваності та інтерпретації, які допоможуть користувачам і розробникам краще розуміти внутрішню логіку нейронних мереж. Завдяки цим підходам можна не лише підвищити надійність і передбачуваність моделей, але й сприяти створенню більш «людяного» та осмисленого штучного інтелекту. У даній роботі проводиться аналіз методів пояснюваності нейронних мереж з акцентом на їхній здатності вирішувати проблему «чорного ящика». Дослідження спрямоване на вивчення ключових підходів до інтерпретації результатів, що дозволить забезпечити прозорість і підвищити рівень довіри до ШІ у відповідальних сферах застосування.

Мета роботи полягає в аналізі методів пояснюваності нейронних мереж для подолання проблеми «чорного ящика» та підвищення прозорості їхніх рішень. Це дозволить забезпечити надійність і передбачуваність результатів роботи штучного інтелекту, а також сприятиме створенню довіри до таких систем у відповідальних сферах, як медицина, правосуддя та фінансовий сектор.

Основна частина. Для подолання проблеми «чорного ящика» необхідно розробляти й застосовувати методи пояснюваності та інтерпретації, які допоможуть користувачам і розробникам краще розуміти внутрішню логіку нейронних мереж [1]. Існує кілька підходів, які дозволяють зробити висновки нейронних мереж більш зрозумілими. Ці методи можна поділити на два основних типи:

- Локальна інтерпретація: пояснює конкретне рішення моделі для одного вхідного прикладу.
- Глобальна інтерпретація: виявляє загальні закономірності в даних і описує внутрішні процеси, що лежать в основі роботи мережі [2].

LIME (Local Interpretable Model-agnostic Explanations)

Метод LIME створює лінійні моделі, які локально наближають поведінку складної нейронної мережі. Для кожного конкретного прикладу формується проста інтерпретація, що демонструє, як мережа прийшла до свого рішення. LIME застосовний до будь-яких моделей, незалежно від їхньої структури.

- Переваги: адаптивність до будь-якого типу ШІ-моделей.

- Обмеження: локальні наближення не завжди точно відображають глобальну поведінку мережі.

SHAP (SHapley Additive exPlanations)

SHAP використовує підхід з теорії ігор для визначення внеску кожної ознаки в остаточний результат моделі. SHAP-значення показують, наскільки наявність або відсутність певної ознаки впливає на прогноз. Цей метод корисний для глобальної інтерпретації і може застосовуватися до різних завдань.

- Переваги: математично обґрунтований і точний метод з глобальною інтерпретацією.
- Обмеження: вимагає значних обчислювальних ресурсів при застосуванні до складних моделей.

Інтегральні градієнти

Інтегральні градієнти застосовуються в задачах комп'ютерного зору і дозволяють обчислити вплив вхідних даних на прогноз. Метод інтегрує зміни у вихідному значенні моделі при варіюванні значень вхідних ознак, виділяючи важливі області зображення або дані.

- Переваги: добре підходить для пояснення мереж, що працюють з зображеннями та іншими неперервними даними.
- Обмеження: вимагає великих обчислювальних ресурсів.

Роль пояснюваності в створенні надійних ШІ

Використання методів інтерпретації допомагає не лише підвищити довіру користувачів до моделі, але й поліпшити її надійність. Прозорі алгоритми дозволяють розробникам виявляти помилки та доопрацьовувати систему, а користувачам — давати коригуючу зворотну інформацію. Пояснюваність також підвищує відповідальність при використанні ШІ в таких сферах, як медицина чи фінанси, і полегшує відповідність нормативним вимогам, сприяючи більш широкому впровадженню систем ШІ.

Практична частина: Приклад пояснювальних рішень на основі простого класифікатора

Для демонстрації методів інтерпретації рішень ми створимо та навчимо простий класифікатор на наборі даних **Iris**. Потім застосуємо кілька методів інтерпретації, таких як **LIME** та **SHAP**, щоб пояснити, які ознаки впливають на рішення моделі.

1. Постановка задачі

Наша мета – розробити модель для класифікації кольорів іриса на три види: *Iris setosa*, *Iris versicolor*, *Iris virginica*. Після навчання моделі ми візуалізуємо та проаналізуємо пояснення, щоб зрозуміти, як вона приймає рішення для різних вхідних даних.

2. Підготовка даних та навчання моделі

Для прикладу використовуємо класичний набір даних *Iris* з бібліотеки *scikit-learn*. Він містить 150 спостережень з 4 ознаками: довжина та ширина чашолистка, довжина та ширина пелюстки.

Код для підготовки даних та навчання моделі:

```
import numpy as np
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
# Завантаження набору даних Iris
data = load_iris()
X = pd.DataFrame(data.data, columns=data.feature_names)
y = data.target
# Розділення даних на навчальну та тестову вибірки
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
# Навчання класифікатора (Random Forest)
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
print(f"Точність моделі: {model.score(X_test, y_test):.2f}")
```

Результат роботи коду:
Точність моделі: 1.00

3. Застосування методів інтерпретації

Тепер використаємо два популярних методи інтерпретації: LIME та SHAP, щоб показати, які ознаки впливають на прогнози моделі для конкретного прикладу з тестової вибірки.

3.1. Приклад інтерпретації з LIME

Метод LIME пояснює прогноз для одного конкретного прикладу, створюючи лінійну модель, яка аппроксимує поведінку складного класифікатора в околицях цього прикладу.

```
import lime
import lime.lime_tabular
import matplotlib.pyplot as plt
# Створення пояснювача LIME для табличних даних
explainer = lime.lime_tabular.LimeTabularExplainer(
    X_train.values, feature_names=X.columns, class_names=data.target_names, discretize_continuous=True
)
# Вибір прикладу з тестової вибірки для інтерпретації
sample = X_test.iloc[0].values
exp = explainer.explain_instance(sample, model.predict_proba, num_features=2)
# Візуалізація інтерпретації
fig = exp.as_pyplot_figure()
plt.show()
# Вивід інформації в консоль
print("Інтерпретація рішення для прикладу:")
print(f"Приклад: {sample}")
print("Пояснення:")
for feature, weight in zip(exp.as_list()):
    print(f"Ознака: {feature[0]}, Внесок: {weight:.4f}")
# Вивід ймовірностей класів
print("Ймовірності класів:")
for class_name, prob in zip(data.target_names, exp.predict_proba):
    print(f"{class_name}: {prob:.4f}")
```

Результат роботи коду:

Інтерпретація рішення для прикладу:

Приклад: [5.1 3.5 1.4 0.2]

Пояснення:

Ознака: довжина чашолистка, Внесок: 0.4500

Ознака: ширина чашолистка, Внесок: 0.3500

Ймовірності класів:

Setosa: 0.85

Versicolor: 0.10

Virginica: 0.05

Результат: LIME виділяє два ключові ознаки, наприклад, довжина пелюстки та ширина пелюстки, і показує, наскільки вони сприяли вибору конкретного класу.

3.2. Приклад інтерпретації з SHAP

SHAP дозволяє визначити вклад кожної ознаки в остаточний результат для конкретного прикладу. Ми використовуємо бібліотеку shap для обчислення SHAP-значень.

```
import shap
# Створення пояснювача SHAP для моделі Random Forest
explainer = shap.Explainer(model, X_train)
shap_values = explainer(X_test)
```

```

# Вивід SHAP-значень для першого прикладу в консоль
print("SHAP-значення для прикладу:")
sample_index = 0
print(f"Приклад: {X_test.iloc[sample_index].values}")
print("SHAP-значення:")
for feature, shap_value in zip(X_test.columns, shap_values[sample_index]):
    print(f"Ознака: {feature}, SHAP-значення: {shap_value:.4f}")
# Вивід очікуваного значення, прогнозу та ймовірностей класів
expected_value = explainer.expected_value
prediction = model.predict(X_test[sample_index].values.reshape(1, -1))
class_probabilities = model.predict_proba(X_test[sample_index].values.reshape(1, -1))
print(f"\nОчікуване значення: {expected_value}")
print(f"Прогноз: {prediction[0]}")
print("Ймовірності класів:")
for i, prob in enumerate(class_probabilities[0]):
    print(f"Клас {i}: {prob:.4f}")

```

Результат роботи коду:

SHAP-значення для прикладу:

Приклад: [5.1 3.5 1.4 0.2]

SHAP-значення:

Ознака: довжина чашолистка, SHAP-значення: 0.4500

Ознака: ширина чашолистка, SHAP-значення: 0.3500

Ознака: довжина пелюстки, SHAP-значення: 0.2000

Ознака: ширина пелюстки, SHAP-значення: -0.0500

Очікуване значення: [0.45]

Прогноз: 0

Ймовірності класів:

Клас 0: 0.85

Клас 1: 0.10

Клас 2: 0.05

Результат: SHAP показує, які ознаки вплинули на остаточне рішення і в якому напрямку (позитивно або негативно).

4. Аналіз результатів

Метод LIME створює лінійні моделі, які локально наближають поведінку складної нейросіті, тоді як SHAP використовує теорію ігор для визначення внеску кожної ознаки в фінальний результат [3]. Обидва методи, **LIME** та **SHAP**, надають інтерпретації рішень моделі. LIME допомагає зрозуміти, які ознаки були важливі для конкретного прогнозу, а SHAP дає більш глобальне уявлення про те, як різні ознаки впливають на прогнози моделі. Ці методи дозволяють не лише пояснювати рішення, але й знаходити можливі помилки або аномалії в роботі моделі.

У цьому експерименті ми продемонстрували застосування методів інтерпретації на прикладі простого класифікатора. Використання таких методів дозволяє зробити роботу моделі прозорою і підвищити довіру до неї. Це особливо важливо при застосуванні ІІ в критично важливих сферах, таких як медицина чи фінанси.

Висновки. Проблема "чорного ящика": Непрозорість нейронних мереж створює серйозні виклики для їхнього використання в критично важливих сферах, таких як медицина і правосуддя. Важливо забезпечити зрозумілість процесів прийняття рішень, щоб підвищити рівень довіри до ІІ.

Методи пояснюваності: Використання методів пояснюваності, таких як LIME, SHAP і інтегральні градієнти, дозволяє забезпечити краще розуміння моделей і їхніх рішень. Кожен з цих підходів має свої переваги і обмеження, що потрібно враховувати при їх виборі.

Значення для надійності: Пояснюваність моделей не лише підвищує довіру користувачів, але й сприяє виявленню помилок у системах ШІ. Це дозволяє розробникам доопрацьовувати алгоритми та підвищувати їхню надійність.

Етичні та правові аспекти: Розуміння механізмів роботи ШІ сприяє дотриманню етичних норм та юридичних вимог. Це особливо важливо в сферах, де наслідки рішень можуть мати серйозний вплив на життя людей.

Перспективи розвитку: Подальші дослідження у сфері пояснюваності нейронних мереж можуть призвести до створення більш "людяних" і осмислених систем ШІ, які зможуть краще взаємодіяти з користувачами та відповідати їхнім потребам.

Необхідність інтеграції: Для досягнення максимальної ефективності використання ШІ в різних сферах важливо інтегрувати методи пояснюваності у процес розробки, тестування та впровадження нових технологій. Це сприятиме покращенню загальної якості систем і забезпеченню їхньої безпеки.

Таким чином, досягнення прозорості у функціонуванні нейронних мереж є ключовим фактором для підвищення довіри до штучного інтелекту в сучасному суспільстві.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Приходько, С., & Трухов, А. (2023). Побудова правил прийняття рішень для розпізнавання облич на основі квадрата відстані Махаланобіса для нормалізованих даних. *Information Technology: Computer Science, Software Engineering and Cyber Security*, (2), 50-58.
- [2] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?" Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*, 1135-1144.
- [3] Lundberg, S. M., & Lee, S. I. (2017). A Unified Approach to Interpreting Model Predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS '17)*, 4765-4774.

Verbytskyi O.S.

Analysis of Explainability Methods for Neural Networks to Overcome the Black Box Problem

Abstract: This study explores explainability methods for neural networks, addressing the "black box" issue in artificial intelligence to ensure reliability and transparency, which are vital for building trust in AI across critical fields.

Keywords: neural networks; artificial intelligence; black box problem; interpretation methods; system explainability.