МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ В. Н. КАРАЗІНА

Кафедра комп`ютерної фізики

"ЗАТВЕРДЖУЮ"

Директор навчально-наукового інституту

комп'ютерної фізики та енергетики

_____ Ірина ГАРЯЧЕВСЬКА

«28» _____ 2024 р.

РОБОЧА ПРОГРАММА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ЗАСТОСУВАННЯ МАШИННОГО НАВЧАННЯ В ЗАДАЧАХ ФІЗИКИ

рівень вищої освіти	другий (магістерський)
галузь знань	10 Природничі науки
спеціальність	105 Прикладна фізика та наноматеріали
освітня програма	«Комп'ютерна фізика»
вид дисципліни	вибіркова

навчально – науковий інститут комп'ютерної фізики та енергетики

2024 / 2025 навчальний рік

Програму рекомендовано до затвердження вченою радою навчально-наукового інституту комп'ютерної фізики та енергетики

28 серпня 2024 року, протокол № 8/24

РОЗРОБНИКИ ПРОГРАМИ:
Немченко Костянтин Едуардович, д.ф.-м.н., професор

Програму схвалено на засіданні кафедри комп'ютерної фізики

Протокол від 28 серпня 2024 року № 8/24

Завідувач кафедри комп'ютерної фізики

_____ Костянтин НЄМЧЕНКО

Програму погоджено з гарантом освітньо-наукової програми «Комп'ютерна фізика»

Гарант освітньо-наукової програми «Комп'ютерна фізика»

_____ Костянтин НЄМЧЕНКО

Програму погоджено науково-методичною комісією
навчально-наукового інституту комп'ютерної фізики та енергетики

Протокол від 28 серпня 2024 року № 8/24

Голова науково-методичної комісії
навчально-наукового інституту комп'ютерної фізики та енергетики

_____ Денис ПРОТЕКТОР

ВСТУП

Програма навчальної дисципліни «Застосування машинного навчання в задачах фізики» складена відповідно до освітньо-професійної програми «Комп'ютерна фізика» підготовки другого (магістерського) рівня вищої освіти

спеціальності (напряму) 105 Прикладна фізика і наноматеріали

1. Опис навчальної дисципліни

Предметом вивчення навчальної дисципліни є використання прикладних пакетів для задач фізики та енергетики.

1.1. Мета викладання навчальної дисципліни

Метою курсу «Застосування машинного навчання в задачах фізики» є вивчення та самостійне користування алгоритмами діагностики зображень або масивів даних з використанням самостійно розроблених алгоритмів та бібліотек для задач фізики та математики.

1.2. Основні завдання вивчення дисципліни

Основним завданням курсу «Застосування машинного навчання в задачах фізики» є застосування математичних алгоритмів для діагностування та керування складними системами керування процесами в фізиці та енергетиці.

У результаті вивчення даного курсу студент повинен

знати: методи діагностики даних за допомогою математичних алгоритмів та самостійне створення програмних продуктів, що реалізують алгоритми автоматизованої діагностики для певних задач в галузі фізики та енергетики.

вміти: застосовувати отримані знання на практиці при діагностиці та аналізу масивів даних за допомогою методів паралельних обчислень, узагальнювати вивчені алгоритмі на складні системи.

Для вивчення курсу необхідні знання з програмування, теорії алгоритмів, математичного аналізу, диференціальних рівнянь, математичної фізики, та курсів лінійної алгебри та аналітичної геометрії.

Основними формами викладання навчального матеріалу з дисципліни «Застосування машинного навчання в задачах фізики» є лекції, лабораторні заняття та самостійна робота студентів.

Методика викладання ґрунтується на проведенні лекційних (2 год/тиждень), та лабораторних (4 год/тиждень) у 2-му семестрі, із загальним аудиторним обсягом 96 годин, що разом з 174 годинами самостійної роботи складає 270 годин загального обсягу. Лекційні заняття проводяться методом семінару. Лабораторні заняття – шляхом розробки окремими студентами завдань програм на мові Python в певному середовищі розробки за вибором студента.

1.3. Кількість кредитів 9

1.4. Загальна кількість годин 270

1.5. Характеристика навчальної дисципліни	
Вибіркова	
Денна форма навчання	Заочна (дистанційна) форма навчання
Рік підготовки	
1-й	-й
Семестр	
2-й	-й
Лекції	
32 год.	год.
Практичні, семінарські заняття	
0 год.	год.
Лабораторні заняття	

64 год.	год.
Самостійна робота	
174 год.	год.
Індивідуальні завдання	
год.	

1.6. Заплановані результати навчання

В результаті вивчення навчальної дисципліни студенти оволодіють засобами застосування математичних алгоритмів для діагностування та керування складними системами керування процесами в фізиці та енергетиці.

1. ПРН1. Використовувати знання в галузі прикладної фізики, математики, електроніки та інформаційних технологій для виконання наукових досліджень та розв'язання виробничих задач.
2. ПРН2. Знаходити та аналізувати наукову та науково-технічну інформацію в галузі прикладної фізики та наноматеріалів із вітчизняних та зарубіжних джерел, в тому числі з використанням сучасних пошукових систем.
3. ПРН4. Встановлювати та аргументувати нові залежності між параметрами та характеристиками фізичних систем.

2. Тематичний план навчальної дисципліни

Розділ 1. Використання мови програмування Python в прикладних задачах

Тема 1. Основні галузі використання в фізиці

Використання Python у фізиці для розв'язання задач у таких галузях, як механіка, термодинаміка, електродинаміка, астрономія, квантова механіка.

Тема 2. Основні правила програмування для розв'язання фізичних задач

Основні принципи програмування для фізичних задач: читаємість коду, коментування, модульність, та організація програм для чисельних обчислень.

Тема 3. Основні бібліотеки

Основні бібліотеки Python для фізичних задач: NumPy (числові обчислення), SciPy (члени складніших розрахунків), SymPy (символьні обчислення), Matplotlib (візуалізація даних).

Тема 4. Структури даних для обробки експериментів

Різні структури даних для роботи з експериментальними даними: списки, масиви, словники, DataFrame (Pandas).

Тема 5. Введення та виведення даних

Механізми введення та виведення даних у Python: зчитування з консолі, виведення на екран, робота з файлами.

Тема 6. ООП для задач фізики

Використання об'єктно-орієнтованого програмування для моделювання фізичних систем через класи та об'єкти.

Тема 7. Нестандартні бібліотеки

Спеціалізовані бібліотеки для фізичних задач: PyDy (динамічні системи), Mayavi (3D візуалізація), QuantumOptics (моделювання квантових систем).

Тема 8. Особливості використання функцій для фізичних задач

Використання функцій для організації та повторного використання коду при розв'язанні фізичних задач.

Розділ 2. Загальна теорія розпізнавання та обробки даних

Тема 1. Задача розпізнавання

Задача розпізнавання включає в себе процес ідентифікації об'єктів або шаблонів за допомогою алгоритмів машинного навчання. Це може бути розпізнавання зображень, тексту, голосу або інших типів даних.

Тема 2. Математична постановка задачі класифікації. Класифікація — це задача віднесення об'єкта до однієї з заданих категорій на основі певних ознак. Математично задача

класифікації формулюється через побудову функції, яка приймає вхідні дані і призначає їх до класу.

Тема 3. Класифікація методів розпізнавання. Створення критеріїв розпізнавання

Методи розпізнавання: статистичні, евристичні, з використанням глибинного навчання тощо. Переваги та недоліки. Критерії розпізнавання, що базуються на ефективності, точності та здатності до навчання на нових даних.

Тема 4.Паралельна та послідовна класифікації.

У паралельній класифікації кілька класифікаторів працюють одночасно, що дозволяє збільшити точність розпізнавання. Послідовна класифікація використовує послідовні етапи для підвищення ефективності.

Тема 5. Основні алгоритми автоматизованої діагностики

Основні алгоритми автоматизованої діагностики включають використання різноманітних технік класифікації, таких як дерева рішень, нейронні мережі, метод опорних векторів (SVM), та інші, для виявлення та діагностики проблем у різних системах.

Тема 6. Критерії вибору алгоритмів.

Вибір алгоритму для розпізнавання чи класифікації базується на таких критеріях, як точність, швидкість виконання, стійкість до шуму в даних, здатність до масштабування та інтерпретованість результатів.

Тема 7. Алгоритми створення моделей

Моделювання включає в себе створення математичних моделей для розпізнавання об'єктів чи патернів. Це може включати використання лінійних моделей, моделей на основі дерев рішень, нейронних мереж тощо.

Тема 8. Автоматична обробка даних

Автоматична обробка даних включає в себе попередню обробку даних, їх очищення, нормалізацію та перетворення в зручний формат для подальшого аналізу та застосування алгоритмів розпізнавання.

3. Структура навчальної дисципліни

Назви розділів	Кількість годин											
	денна форма						заочна форма					
	усього	у тому числі					усього	у тому числі				
		л	п	лаб.	інд.	с. р.		л	п	лаб.	інд.	с. р.
1	2	3	4	5	6	7	8	9	10	11	12	13
Розділ 1. Стандартні бібліотеки шаблонів												
Тема 1	16	2		4		10						
Тема 2	16	2		4		10						
Тема 3	16	2		4		10						
Тема 4	16	2		4		10						
Тема 5	16	2		4		10						
Тема 6	18	2		4		12						
Тема 7	18	2		4		12						
Тема 8	18	2		4		12						
Разом за розділом 1	134	16		32		86						
Розділ 2. Паралельні обчислення												
Тема 1	16	2		4		10						
Тема 2	16	2		4		10						
Тема 3	16	2		4		10						
Тема 4	16	2		4		10						

Тема 5	18	2		4		12						
Тема 6	18	2		4		12						
Тема 7	18	2		4		12						
Тема 8	18	2		4		12						
Разом за розділом 2	136	16		32		88						
Усього годин	270	32		64		174						

4. Теми практичних занять

Не передбачені навчальним планом

Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	Основні галузі використання в фізиці	4
2	Основні правила програмування для розв’язання фізичних задач	4
3	Основні бібліотеки	4
4	Структури даних для обробки експериментів	4
5	Введення та виведення даних	4
6	ООП для задач фізики	4
7	Нестандартні бібліотеки	4
8	Особливості використання функцій для фізичних задач	4
9	Задача розпізнавання	4
10	Математична постановка задачі класифікації	4
11	Класифікація методів розпізнавання.	4
12	Паралельна та послідовна класифікації.	4
13	Основні алгоритми автоматизованої діагностики	4
14	Критерії вибору алгоритмів.	4
15	Алгоритми створення моделей	4
16	Автоматична обробка даних	4
	Разом	64

5. Завдання для самостійної робота

№ з/п	Назва теми	Кількість годин
1	Ознайомитись з основним галузями використання в фізиці	10
2	Вивчити основні правила програмування для розв’язання фізичних задач	10
3	Засвоїти основні бібліотеки	10
4	Вивчити структури даних для обробки експериментів	10
5	Розв’язати приклади введення та виведення даних	10
6	Ознайомитись з методами ООП для задач фізики	12
7	Засвоїти нестандартні бібліотеки	12
8	Вивчити особливості використання функцій для фізичних задач	12
9	Розв’язати приклади задач розпізнавання	10
10	Засвоїти математичну постановку задачі класифікації	10
11	Вивчити класифікація методів розпізнавання.	10
12	Провести за прикладом паралельну та послідовну класифікації.	10
13	Розв’язати приклади з використанням основних алгоритмів автоматизованої діагностики	12
14	Ознайомитись з критеріями вибору алгоритмів.	12
15	Вивчити алгоритми створення моделей	12
16	Розв’язати приклади з автоматичної обробки даних	12
	Разом	174

6. Індивідуальні завдання

7. Методи контролю

Навчальна програма нормативної дисципліни «Застосування машинного навчання в задачах фізики» відповідає чинним нормативним документам, рекомендованим Міністерством освіти і науки України як навчальна програма для студентів фізичного спрямування для вищих навчальних закладів. Нормативна навчальна дисципліна «Застосування машинного навчання в задачах фізики» є складовою циклу професійної підготовки фахівців освітньо–кваліфікаційного рівня «магістр».

Зміст програми, об'єм учбових питань дисципліни «Застосування машинного навчання в задачах фізики» визначаються потребою загальнонаукової, загально-інженерної та технічної підготовки.

Послідовність викладання тем не є суворо обов'язковою і може бути змінена у відповідності із структурно-логічною схемою викладання інших дисциплін.

Вивчення дисципліни «Застосування машинного навчання в задачах фізики» передбачає викладання лекцій, проведення практичних занять, виконання лабораторних робіт. Підсумковий контроль знань здійснюється на екзамені.

На лекціях викладається теоретичний матеріал, який ілюструється типовими прикладами і задачами за профілем підготовки фахівців з прикладної фізики та наноматеріалів. Викладання лекційного матеріалу повинно має закінчений характер, здійснюється у доступній і наочній формі, містить проблемні ситуації.

Основною метою практичних занять є розвиток навичок практичного застосування вищої математики і закріплення теоретичного матеріалу. При вирішенні задач рекомендується користуватися стандартними прийомами і методиками. На кожному практичному занятті частину учбового часу доцільно використовувати для самостійного розв'язку задач, контролюючи при цьому правильність обраного методу і ходу рішення.

Важливим фактором засвоєння курсу «Застосування машинного навчання в задачах фізики» й оволодіння її методами є самостійна робота студентів. Для самостійного відпрацювання розділів і тем дисципліни пропонуються лабораторні роботи, при перевірці виконання яких здійснюється ефективний контроль за рівнем засвоєння матеріалу.

На всіх заняттях підкреслюється роль і значення предмету на конкретних прикладах і задачах прикладного характеру, висвітлюється її зв'язок з іншими дисциплінами і майбутньою професією.

8. Схема нарахування балів

Поточний контроль, самостійна робота, індивідуальні завдання	
Розділ 1	
Т1	2
Т2	2
Т3	2
Т4	2
Т5	2
Т6	2
Т7	2
Т8	2
Контрольна робота	14
Розділ 2	
Т1	2
Т2	2
Т3	2
Т4	2
Т5	2
Т6	2
Т7	2

Т8	2
Контрольна робота	14
Екзаменаційна робота	40
Сума	100

Т1, Т2 … Т8 – теми розділів.

Мінімальна кількість балів для допуску студента до підсумкового іспиту становить – 15 балів.

Критерії оцінювання

З кожну тему - 6(7) балів дається одне завдання (задача), розв'язок якої оцінюється за такими критеріями:

Наявність відповіді – 1 бал
Коректність викладок – 2(1) бали
Логічність викладок – 1 бал
Коректність визначень – 1 бал
Повнота відповіді – 1 бал
Знання цілей задачі – 1 бал

Шкала оцінювання

Сума балів за всі види навчальної діяльності протягом семестру	Оцінка	
	для екзамену	для заліку
90 – 100	відмінно	зараховано
70-89	добре	
50-69	задовільно	
1-49	незадовільно	не зараховано

9. Рекомендована література

Основна література

1. Ефективне використання STL Скотт Мейєрс 4.5 ISBN: 5-94723-382-7 2002
2. C++ та STL. Девід Р. Мюссер, Жилмер Дж. Дердж, Атул Сейні 5.0 ISBN: 978-5-8459-1665-5, 978-0-321-70212-8: 2010
3. Розширення бібліотеки STL для C++. Набори та ітератори (+ CD… Метью Вілсон 0.0 ISBN: 978-5-94074-442-9, 978-5-9775-0196-5, 978-0-321-30550-7 2008
4. Узагальнене програмування та STL. Використання та нарощування… Метью Г. Остерн 5.0 ISBN: 5-7940-0119-4, 0-201-30956-4 2004

Допоміжна література

1. STL. Кишеньковий довідник Рей Лішнер 0.0 ISBN: 5-469-00389-2, 0596005563 2005
2. Стандартна бібліотека C++. Ніколаї М. Джосаттіс 4.5 ISBN: 978-5-8459-1837-6 2014

10. Посиланная на інформаційні ресурси в Інтернеті, відео-лекції, інше методичне забезпечення

1. http://www.rsdn.ru/res/book/cpp/josattis.xml
2. http://www.rsdn.ru/res/book/cpp/halpern.xml

Додаток до робочої програми

Пояснення за темами з прикладами лабораторних завдань

Тема 1. Задача розпізнавання
Пояснення: Розпізнавання полягає в ідентифікації об'єктів, образів або даних з набору можливих варіантів. Це включає в себе класифікацію та виявлення патернів на основі наявних даних.
Приклад:
Задача розпізнавання може бути такою: маємо набір зображень цифр від 0 до 9, і необхідно автоматично розпізнати, яка цифра зображена на кожному зображенні.
Практичне завдання:
Використати бібліотеку **scikit-learn** для розпізнавання рукописних цифр за допомогою датасету MNIST. Це популярний набір даних, який містить 28x28 пікселів для кожного зображення цифр від 0 до 9.

```
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import classification_report

# Завантажуємо набір даних
digits = load_digits()

# Розділяємо дані на навчальну та тестову вибірки
X_train, X_test, y_train, y_test = train_test_split(digits.data, digits.target, test_size=0.3,
random_state=42)

# Створення та навчання моделі
model = SVC(gamma=0.001)
model.fit(X_train, y_train)

# Оцінка моделі
y_pred = model.predict(X_test)
print(classification_report(y_test, y_pred))
```

Тема 2. Математична постановка задачі класифікації
Пояснення: Класифікація — це задача віднесення елемента до одного з попередньо визначених класів. Математично задача класифікації формується як побудова функції f(x)f(x), яка відносить вхідні дані xx до одного з класів yy.

Приклад:

Маємо набір даних, де кожен елемент має кілька ознак (наприклад, вага, вік) та потрібно віднести його до класу: "чоловік" або "жінка".

Практичне завдання:

Завантажити набір даних, що містить інформацію про людей, та побудувати модель класифікації, що визначає стать людини на основі таких ознак, як вік та зріст.

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score

# Завантажуємо набір даних Iris
iris = load_iris()

# Розділяємо дані на навчальні та тестові
X_train, X_test, y_train, y_test = train_test_split(iris.data, iris.target, test_size=0.3,
random_state=42)
```

```
# Створюємо та навчаємо модель
model = DecisionTreeClassifier()
model.fit(X_train, y_train)

# Оцінка точності
y_pred = model.predict(X_test)
print(f"Точність моделі: {accuracy_score(y_test, y_pred)}")
```

Тема 3. Класифікація методів розпізнавання. Створення критеріїв розпізнавання
Пояснення: Існують різні методи класифікації, включаючи статистичні методи, методи на основі правил, нейронні мережі та інші. Критерії розпізнавання включають точність, швидкість, стійкість до шуму тощо.

Приклад:

- **Лінійний дискримінантний аналіз (LDA)** — метод, що розділяє класи за допомогою лінійних меж.
- **Метод опорних векторів (SVM)** — метод, який знаходить гіперплощину для поділу класів.

Практичне завдання:

Використати **SVM** для класифікації тексту або зображень. Можна спробувати класифікувати зображення рукописних цифр із набору MNIST.

```
from sklearn.svm import SVC

# Створення моделі SVM та навчання
model_svm = SVC(kernel='linear')
model_svm.fit(X_train, y_train)

# Прогнозування
y_pred_svm = model_svm.predict(X_test)
print(f"Точність моделі SVM: {accuracy_score(y_test, y_pred_svm)}")
```

Тема 4. Паралельна та послідовна класифікації
Пояснення: Паралельна класифікація передбачає одночасне використання кількох класифікаторів, тоді як послідовна класифікація здійснюється поетапно, кожен етап залежить від попереднього.

Приклад:

- **Паралельна класифікація**: кілька алгоритмів, наприклад, дерево рішень та SVM, працюють одночасно, і результати комбінуються.
- **Послідовна класифікація**: один класифікатор використовує результати іншого для підвищення точності.

Практичне завдання:

Спробувати комбінувати різні класифікатори, наприклад, використовуючи **VotingClassifier** для паралельної класифікації.

```
from sklearn.ensemble import VotingClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression

# Створення комбінованої моделі
knn = KNeighborsClassifier()
logreg = LogisticRegression()
voting_clf = VotingClassifier(estimators=[('knn', knn), ('logreg', logreg)], voting='hard')
```

```
# Навчання та оцінка
voting_clf.fit(X_train, y_train)
y_pred_voting = voting_clf.predict(X_test)
print(f"Точність VotingClassifier: {accuracy_score(y_test, y_pred_voting)}")
```

Тема 5. Основні алгоритми автоматизованої діагностики

Пояснення: Для автоматизованої діагностики використовуються алгоритми, які можуть виявити аномалії або помилки в системах. Це включає деревоподібні структури, нейронні мережі та методи статистичної класифікації.

Приклад:

- **Дерева рішень**: використовуються для виявлення проблем у діагностичних даних.
- **Нейронні мережі**: застосовуються для складних задач діагностики, таких як медичні зображення.

Практичне завдання:

Створити модель, яка виявляє можливі проблеми в даних. Наприклад, за допомогою алгоритму дерева рішень виявляти хвороби на основі симптомів.

Тема 6. Критерії вибору алгоритмів

Пояснення: Вибір алгоритму залежить від різних критеріїв: точності, швидкості, стійкості до шуму, масштабованості. Наприклад, для малих даних можна використовувати логістичну регресію, а для великих — SVM або нейронні мережі.

Практичне завдання:

Порівняти різні алгоритми для вирішення однієї задачі та вибрати найкращий за точністю та швидкістю. Наприклад, порівняти **Random Forest** та **SVM** на задачі класифікації.

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC

# Створення моделей
rf_model = RandomForestClassifier(n_estimators=100)
svm_model = SVC(kernel='linear')

# Навчання та оцінка
rf_model.fit(X_train, y_train)
svm_model.fit(X_train, y_train)

# Оцінка точності
print(f"Точність RandomForest: {accuracy_score(y_test, rf_model.predict(X_test))}")
print(f"Точність SVM: {accuracy_score(y_test, svm_model.predict(X_test))}")
```

Тема 7. Алгоритми створення моделей

Пояснення: Моделювання полягає в побудові математичних або статистичних моделей, які можуть описати поведінку системи або явища на основі даних.

Приклад:

- **Регресія**: для прогнозування числових значень.
- **Класифікація**: для віднесення до категорій.

Пра

ктичне завдання: Створити модель для прогнозування ціни житла на основі таких факторів, як площа, кількість кімнат, розташування.

Тема 8. Автоматична обробка даних

Пояснення: Автоматична обробка даних включає попереднє очищення, нормалізацію, перетворення та підготовку даних для подальшого аналізу.

Приклад:

- **Нормалізація даних**: приведення всіх змінних до одного діапазону.
- **Обробка пропущених значень**: заповнення або видалення пропусків.

Практичне завдання:

Виконати нормалізацію даних та очистити пропуски за допомогою бібліотеки **Pandas**.

```
import pandas as pd
from sklearn.preprocessing import StandardScaler

# Завантажуємо дані
data = pd.read_csv("data.csv")

# Заповнюємо пропуски середнім значенням
data.fillna(data.mean(), inplace=True)

# Нормалізація даних
scaler = StandardScaler()
data_normalized = scaler.fit_transform(data)
```