**Звіт до комп’ютерного практикуму №1.**

**Моделі машинного навчання**

**ПІБ:** Шляхтун Денис Михайлович.

**Група:** ІП-14.

**Мета роботи:** ознайомитись з принципами функціонування, створення, навчання та використання моделей машинного навчання.

**Завдання:** для обраної задачі класифікації (або регресії) на основі типового датасету створити модель машинного навчання, навчити її на датасеті, перевірити результат на тестовій вибірці.

**Номер варіанту:** 22.

**Завдання для варіанту:** Fashion MNIST, тришаровий персептрон.

**Засоби виконання практикуму:** Visual Studio Code, Python, Tensorflow

**Набір даних (датасет):** Fashion MNIST - це набір даних зображень товарів від Zalando, що складається з навчального набору з 60 000 прикладів і тестового набору з 10 000 прикладів. Кожен приклад - це зображення розміром 28х28 пікселів у відтінках сірого, пов'язане з міткою з 10 класів.

Зображення, що містить текст, білий, ескіз, чорно-білий

Автоматично згенерований опис

**Задача:** класифікація.

**Попередня обробка даних:** перетворено значення пікселів від [0; 255] до [0; 1] – здійснено нормалізацію даних.

**Модель машинного навчання:** Багатошаровий персептрон, або мережа прямого поширення – архітектура нейронної мережі з послідовними з’єднаннями шарів: передають інформацію від входу до виходу (немає зворотніх зв’язків); складаються з вхідних, прихованих або вихідних нейронів, згрупованих в шари; нейрони одного шару не пов'язані між собою, а сусідні шари зазвичай повністю пов'язані.

Тришаровий персептрон складається з вхідного шару, двох прихованих шарів та вихідного шару.

На вхід надаються дані у формі (28, 28, 1) – розмір картинки та кількість кольорів і за допомогою Flatten() формують єдиний вектор для подальшої роботи. Кількість виходів обмежується кількістю класів.

def simple\_mlp\_model(num\_classes):

input\_ = tf.keras.layers.Input(shape=(28, 28, 1,))

x = tf.keras.layers.Flatten()(input\_)

x = tf.keras.layers.Dense(256, activation='relu')(x)

x = tf.keras.layers.Dense(128, activation='relu')(x)

output\_ = tf.keras.layers.Dense(num\_classes, activation='softmax')(x)

return tf.keras.models.Model(input\_, output\_, name='Classifier')

num\_classes = metadata.features['label'].num\_classes

model = simple\_mlp\_model(num\_classes)

model.summary()

**Зображення, що містить текст, знімок екрана, програмне забезпечення, схема

Автоматично згенерований опис**

**Навчання моделі:** Спочатку здійснюється компіляція моделі, для якої потрібно вказати оптимізатор, функцію витрат та метрику. Обрано оптимізатор Adam як адаптивну варіацію стохастичного градієнтного спуску, що має поширення у використанні у нейронних мережах, бо є відносно швидкою та ефективною у використанні пам’яті. Функція втрат SparseCategoricalCrossentropy() була використана, бо стоїть задача класифікації. Метрика accuracy є оптимальною для загальної правильності визначення класів.

Навчання здійснювалося за допомогою методу fit(), у якому вказуються тренувальні дані, кількість епох – 20 та валідаційні дані.

**Результати навчання:** числові результати та динаміка навчання, бажано наочно представити результати.

Перший Dense шар – 64 нейрони, другий Dense шар – 32 нейрони.

Accuracy: 0.9006

Loss: 0.2771

Validation accuracy: 0.8917

Validation loss: 0.3177

Test accuracy: 0.8841

Test loss: 0.3232

Зображення, що містить текст, знімок екрана, ряд, Графік

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, Графік, схема

Автоматично згенерований опис

Перший Dense шар – 256 нейронів, другий Dense шар – 128 нейронів.

Accuracy: 0.9316

Loss: 0.1893

Validation accuracy: 0.8965

Validation loss: 0.3155

Test accuracy: 0.8895

Test loss: 0.3128

Зображення, що містить текст, знімок екрана, Графік, схема

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, Графік, схема

Автоматично згенерований опис

Перший Dense шар – 256 нейронів, шар Dropout(0.5), Dense шар – 128 нейронів.

Accuracy: 0.8901

Loss: 0.3024

Validation accuracy: 0.8913

Validation loss: 0.2927

Test accuracy: 0.8958

Test loss: 0.2857

Зображення, що містить текст, знімок екрана, Графік, схема

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, Графік, ряд

Автоматично згенерований опис

**Оцінка результатів навчання:** Модель вдалося навчити, адже вона розпізнає тип одягу з точністю щонайменше 88% на тестових даних, це не ідеальний результат, але задача не є такою, що вимагає високу точність, наприклад медицина.

Перенавчання майже відсутнє для першого варіанту нейронної мережі з різницею між навчальною і валідаційною точністю менше 1%, більш присутнє у другому варіанті з різницею у понад 3%, для боротьби з перенавчанням третій варіант нейронної мережі є модифікацією другого з додаванням шару Dropout(0.5), очікувано впала навчальна точність і валідаційна точність теж незначно зменшилася. Загалом для усіх трьох варіантів нейронної мережі валідаційна точність майже однакова, але тестова виявилася найкращою для третього.

Точність моделі є оптимальною, тобто вона в деякій мірі підходить під задачу, але скоріше за все є кращі моделі для вирішення цієї задачі.

Для покращення результатів було обрано різні кількості нейронів шарів та додано шар Dropout(0.5) для уникнення перенавчання, але загалом точність трьох моделей на тестових даних відрізняється незначно, тому можливо вияснити лише більшою кількістю експериментів та зміною підходів, як збільшити точність моделі.