

Звіт до комп'ютерного практикуму №2.

Згорткові нейронні мережі

ПІБ: Грисюк Михайло Олександрович

Група: ІК-21мп

Мета роботи: ознайомитись з принципами побудови, навчання та використання згорткових нейронних мереж, дослідити вплив параметрів моделі, алгоритму навчання та даних на результати роботи.

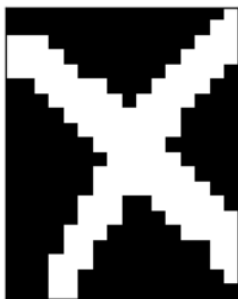
Завдання: для задачі класифікації зображень на основі типового датасету створити згорткову нейронну мережу. Навчити її, перевірити результат на тестовій вибірці, оцінити результати. Провести дослідження впливу параметрів на результати роботи мережі.

Номер варіанту: 5

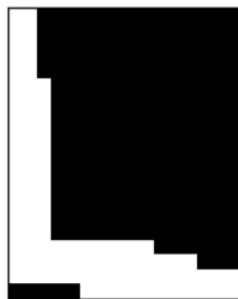
Завдання для варіанту: Для задачі класифікації зображень на основі типового датасету binary alpha digits створити згорткову нейронну мережу. Навчити її, перевірити результат на тестовій вибірці, оцінити результати. Провести дослідження впливу кількості та порядку шарів згортки та пулінгу в базовій CNN на результати роботи мережі.

Засоби виконання практикуму: Дану комп'ютерну практику було виконано в середовищі VSCode зі спеціально встановленими розширеннями Jupiter та іншими. Використовувалась мова програмування Python та фреймворком TensorFlow, це є самі популярні інструменти для створення та навчання нейронних мереж.

Набір даних (датасет): Датасет *binary_alpha_digits* складаються із 1404 прикладів. Це є двійкова 20x16 матриця (зображення) цифр від "0" до "9" і великих "A" до "Z". 39 прикладів кожного класу.



X (33)



L (21)



G (16)

Попередня обробка даних: немає

Модель машинного навчання:

На вході у нас буде матриця 20x16 з бінарними значеннями (0,1), також нам знадобиться внутрішній шар із згортковою нейронною мережею (Conv2D), далі буде шар класифікації і на виході буде 36 класів. Виходить така конструкція мережі:



Код моделі навчання:

```
def simple_convolution_model(num_classes):
    input_ = tf.keras.layers.Input(shape=(20, 16, 1))
    x = tf.keras.layers.Conv2D(32, (3, 3), padding='same', activation='relu')(input_)
    # x = tf.keras.layers.MaxPool2D(2, 2)(x)
    # x = tf.keras.layers.Conv2D(64, (3, 3), padding='same', activation='relu')(x)
    # x = tf.keras.layers.MaxPool2D(2, 2)(x)
    x = tf.keras.layers.Flatten()(x)
    output_ = tf.keras.layers.Dense(num_classes, activation='softmax')(x)

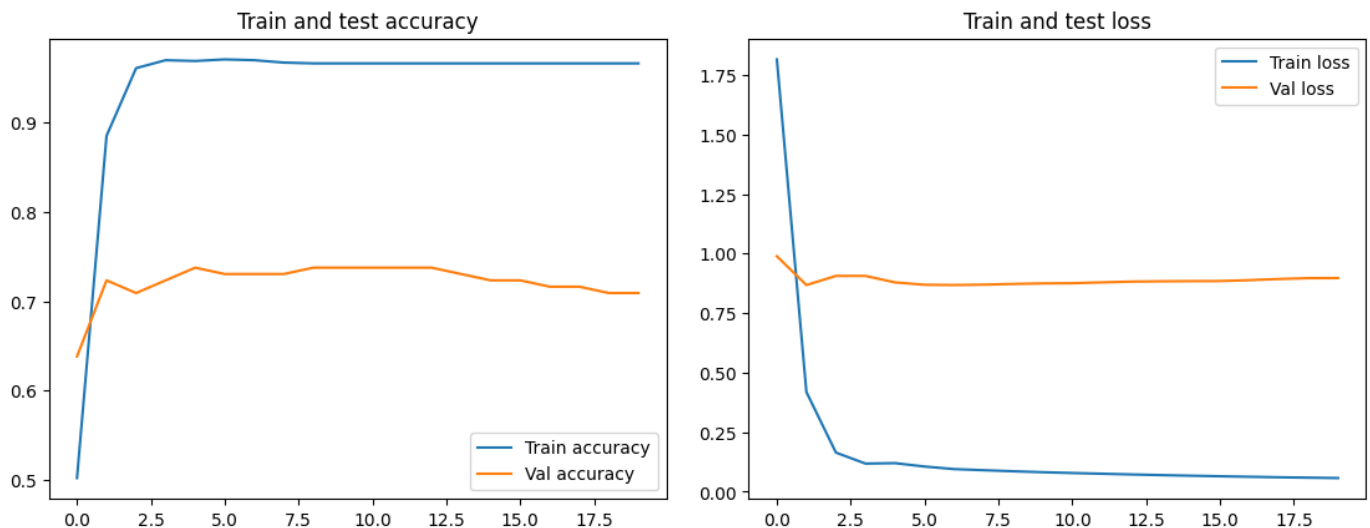
    return tf.keras.models.Model(input_, output_, name='Classifier')
```

Константи:

```
BATCH_SIZE = 4
EPOCHS = 20
LEARNING_RATE = 0.001
✓ 0.2s
```

Навчання моделі: Тут ми використовували модель згорткової мережі. Ми використовували алгоритм навчання мультикласової класифікації, адже нам потрібно знайти до якого класу відноситься даний набір значень, а для функції втрат тут доречно використати **Categorical Crossentropy** - типовий вибір і він вимагає softmax ФА, що підходить для нас. Softmax ми використовували для отримання ймовірностей які вказують до якого класу найбільш відноситься даний набір значень.

Результати навчання:

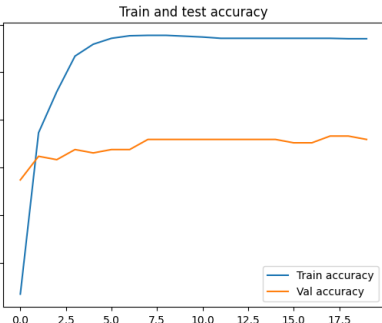
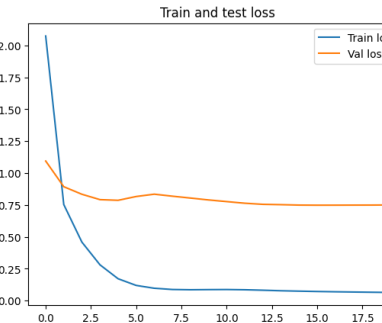
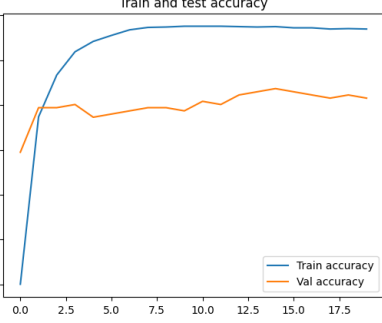
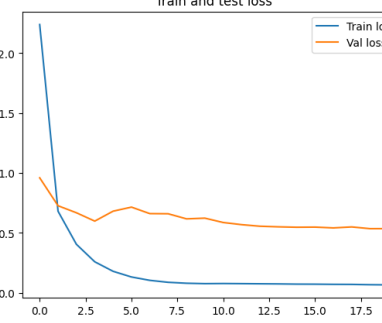
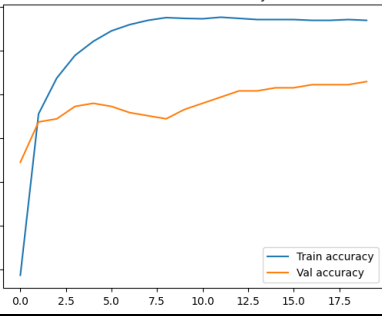
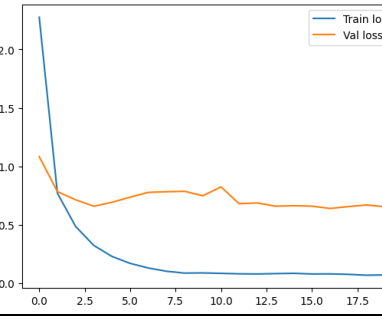


```
35/35 [=====] - 0s 3ms/step - loss: 0.9996 - accuracy: 0.7071
Test loss : 0.9996270537376404
Test Accuracy : 0.7071428298950195
```

Оцінка результатів навчання: Як ми бачимо наша модель навчається, але в нас виходить низька точність та високі втрати. І модель перенавчається. Щоб покращити результат нам можна збільшити кількість та змінити порядок шарів згортки та пулінгу в базовій CNN.

Задача дослідження: Вплив кількості та порядку шарів згортки та пулінгу в базовій CNN.

Результати експериментального дослідження:

Варіант покращення результату	Точність	Втрати	Оцінка результату
Додавання шару MaxPool			train_loss: 0.0643 train_accuracy: 0.9706 val_loss: 0.7501 val_accuracy: 0.7589 test_loss: 0.96 test_accuracy : 0.75 Незначне покращення результату
Додавання шарів (MaxPool-Conv2D(64)-MaxPool)			train_loss: 0.0662 train_accuracy: 0.9697 val_loss: 0.5337 val_accuracy: 0.8156 test_loss : 0.75 test_accuracy : 0.801 Найкраще покращення результату
Змінюєм порядок шарів Conv2D			train_loss: 0.0708 train_accuracy: 0.9697 val_loss: 0.65 val_accuracy: 0.8227 test_loss : 0.993 test_accuracy : 0.771 Значне покращення результату.

Висновки за результатами дослідження: Згідно нашими результатами можна зробити висновок, що при збільшенні кількості та змїну порядку шарів згортки та пулінгу результати точності та втрат покращуються. При подальшому збільшенні шарів результати не будуть покращуватися і буде відбуватись перенавчання. Щоб ще покращити результат потрібно збільшити кількість навчальних прикладів. Краще за все буде використання такої моделі внутрішнього шару:

Conv2D(32, (3,3), Relu) => MaxPool(2, 2) => Conv2D(64, (3,3), Relu) => MaxPool(2, 2)