

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО  
ОБРАЗОВАНИЯ  
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНОЛОГИЧЕСКИЙ  
УНИВЕРСИТЕТ «МИСиС»

---

Институт ИТАСУ  
Группа: МПИ-20-4-2

## ОТЧЕТ

по лабораторной работе №4  
по курсу «Нейронные сети»

Выполнил: Хабибулин М.И.  
группа МПИ-20-4-2  
Проверил: Курочкин И.И.

Москва 2020

---

## ЛР4. Сверточные нейронные сети

### Результаты

- 1) Описание архитектуры сверточной нейронной сети, параметры слоев и настройки обучения (к примеру, типы слоев, количество нейронов в каждом слое, функции активации, начальные значения весов).

Сверточная нейронная сеть состоит из сверточных слоев, и слоев MaxPooling. Также включен Dropout слой для избежания переобучения. На выходе сети добавлен полносвязный слой (Dense), за которым следует слой softmax. Сначала в модель добавляем сверточные слои с 32 фильтрами с размером окна  $3 \times 3$ . Далее добавляется сверточный слой с 64 фильтрами. За каждым слоем добавлен слой максимального пуллинга с размером окна  $2 \times 2$ . Также добавлены слои Dropout с коэффициентами 0,25 и 0.5 для того чтобы не произошло переобучение сети. В заключительных строках добавлен плотный слой Dense, который выполняет классификацию среди 10 а затем 14 классов с использованием функции активации softmax.

Model: "sequential\_3"

Layer (type)	Output Shape	Param #
=====		
conv2d_9 (Conv2D)	(None, 32, 32, 32)	896
activation_13 (Activation)	(None, 32, 32, 32)	0
conv2d_10 (Conv2D)	(None, 30, 30, 32)	9248
activation_14 (Activation)	(None, 30, 30, 32)	0
max_pooling2d_5 (MaxPooling2D)	(None, 15, 15, 32)	0
dropout_7 (Dropout)	(None, 15, 15, 32)	0
conv2d_11 (Conv2D)	(None, 15, 15, 64)	18496
activation_15 (Activation)	(None, 15, 15, 64)	0
conv2d_12 (Conv2D)	(None, 13, 13, 64)	36928
activation_16 (Activation)	(None, 13, 13, 64)	0
max_pooling2d_6 (MaxPooling2D)	(None, 6, 6, 64)	0
dropout_8 (Dropout)	(None, 6, 6, 64)	0
flatten_3 (Flatten)	(None, 2304)	0
dense_11 (Dense)	(None, 512)	1180160
activation_17 (Activation)	(None, 512)	0
dropout_9 (Dropout)	(None, 512)	0
dense_12 (Dense)	(None, 14)	7182
activation_18 (Activation)	(None, 14)	0
=====		
Total params: 1,252,910		
Trainable params: 1,252,910		
Non-trainable params: 0		

#### Описание архитектуры сверточной нейронной сети

## 2. Метрики качества: Accuracy, Precision, Recall, F1-measure, матрица ошибок (confusion matrix)

### Test results

Accuracy	0.6789
loss	0.91403
F1-measure	0,68
Recall	0,68
Precision	0,66

### матрица ошибок (confusion matrix)

```
confusion_matrix(np.argmax(y_test,axis=1), np.argmax(model.predict(x_test), axis=1))
```

```
array([[737, 16, 82, 9, 31, 8, 9, 16, 71, 21],
       [ 45, 771, 13, 10, 11, 3, 11, 11, 35, 90],
       [ 62, 3, 573, 57, 144, 58, 48, 35, 16, 4],
       [ 16, 2, 126, 460, 107, 149, 58, 53, 20, 9],
       [ 21, 2, 84, 45, 681, 19, 37, 97, 13, 1],
       [ 13, 2, 107, 166, 65, 555, 19, 68, 3, 2],
       [ 5, 3, 70, 58, 92, 25, 732, 10, 4, 1],
       [ 14, 1, 53, 31, 86, 57, 2, 750, 3, 3],
       [ 87, 27, 24, 13, 15, 5, 8, 3, 802, 16],
       [ 49, 82, 12, 18, 15, 7, 18, 34, 37, 728]])
```

### Train results

Accuracy	0.70837
loss	0.84357
F1-measure	0,71
Recall	0,70
Precision	0,71

матрица ошибок (confusion matrix)

```
: confusion_matrix(np.argmax(y_train,axis=1), np.argmax(model.predict(x_train),
```

```
: array([[3756, 74, 395, 52, 142, 28, 22, 72, 357, 102],
         [ 159, 4061, 64, 34, 56, 12, 49, 42, 162, 361],
         [ 259, 15, 3125, 237, 664, 223, 209, 174, 75, 19],
         [ 88, 14, 571, 2558, 460, 705, 286, 213, 71, 34],
         [ 123, 5, 377, 169, 3612, 101, 145, 429, 31, 8],
         [ 32, 6, 470, 778, 371, 2879, 94, 332, 22, 16],
         [ 28, 20, 359, 292, 443, 84, 3685, 36, 33, 20],
         [ 47, 8, 193, 176, 448, 218, 25, 3828, 29, 28],
         [ 346, 112, 135, 48, 58, 25, 25, 26, 4137, 88],
         [ 217, 415, 68, 86, 61, 24, 55, 154, 142, 3778]])
```

Сравнение с результатами полученными на:

<https://www.tensorflow.org/tutorials/images/cnn>

loss: 0.8497 - accuracy: 0.7167

- Используя ту же CNN и полученный набор весов провести дообучить CNN на классах из одного суперкласса ([по вариантам](#)) эталонного датасета CIFAR-100. Задача классификации решается на классах CIFAR-10 + классы из одного суперкласса CIFAR-100.

14. Хабибулин Марат Ильдарович non-insect invertebrates crab: lobster, snail, spider, worm

Из достоверных источников известно, что эти четыре класса находятся под номерами 10, 11, 12, 13. Выдернем их из выборки cifar100 и соединим с cifar10. (Смотри код).

Также для ускорения обучения заморозим первые 5ть слоев обученной нейронной сети.

```
: for layer in model.layers[:5]:  
    layer.trainable = False
```

## Результаты для объединенных выборок:

Метрики качества: Accuracy, Precision, Recall, F1-measure, матрица ошибок (confusion matrix)

Test results

Accuracy	0.72769
loss	0.7688
F1-measure	0,73
Recall	0,73
Precision	0,74

матрица ошибок (confusion matrix)

```
: confusion_matrix(np.argmax(y4_test,axis=1), np.argmax(model.predict(x4_test), a  
:  
: array([[756, 26, 66, 21, 9, 12, 9, 12, 55, 34, 0, 0, 0,  
0],  
[ 12, 872, 6, 15, 1, 6, 10, 2, 15, 61, 0, 0, 0,  
0],  
[ 67, 8, 604, 68, 73, 80, 55, 31, 8, 6, 0, 0, 0,  
0],  
[ 12, 3, 77, 560, 32, 217, 60, 19, 9, 11, 0, 0, 0,  
0],  
[ 21, 5, 84, 79, 614, 76, 39, 70, 11, 1, 0, 0, 0,  
0],  
[ 8, 3, 51, 160, 29, 701, 8, 37, 3, 0, 0, 0, 0,  
0],  
[ 4, 5, 50, 70, 26, 40, 794, 7, 2, 2, 0, 0, 0,  
0],  
[ 13, 1, 46, 30, 38, 101, 3, 765, 1, 2, 0, 0, 0,  
0],  
[ 63, 41, 14, 21, 0, 10, 6, 2, 819, 24, 0, 0, 0,  
0],  
[ 26, 88, 9, 25, 4, 12, 9, 7, 21, 799, 0, 0, 0,  
0],  
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 71, 27, 1,  
1],  
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 18, 78, 2,  
2],  
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 7, 83,  
6],  
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 7, 17, 24,  
52]])
```

Train\_results

Accuracy	0.76542
loss	0.67712
F1-measure	0,75
Recall	0,75
Precision	0,76

матрица ошибок (confusion matrix)

```
: confusion_matrix(np.argmax(y4_train,axis=1), np.argmax(model.predict(x4_train), axis=1))
: array([[3969, 119, 311, 85, 48, 38, 23, 51, 225, 131, 0,
        0, 0, 0],
       [ 59, 4552, 34, 34, 10, 30, 16, 17, 53, 195, 0,
        0, 0, 0],
       [ 273, 16, 3300, 339, 304, 336, 254, 121, 41, 16, 0,
        0, 0, 0],
       [ 66, 15, 284, 3083, 126, 1052, 225, 86, 42, 21, 0,
        0, 0, 0],
       [ 118, 8, 367, 312, 3336, 303, 192, 324, 29, 11, 0,
        0, 0, 0],
       [ 19, 5, 174, 652, 141, 3775, 72, 140, 11, 11, 0,
        0, 0, 0],
       [ 21, 28, 245, 311, 109, 157, 4088, 13, 14, 14, 0,
        0, 0, 0],
       [ 34, 11, 153, 211, 184, 444, 14, 3910, 14, 25, 0,
        0, 0, 0],
       [ 263, 148, 88, 89, 17, 32, 24, 14, 4235, 90, 0,
        0, 0, 0],
       [ 119, 398, 31, 88, 4, 55, 24, 57, 73, 4151, 0,
        0, 0, 0],
       [ 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 375,
        104, 5, 15],
       [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 84,
        400, 4, 12],
       [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 47,
        45, 372, 36],
       [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 66,
        75, 103, 256]])
```

### 3. Динамика обучения (Accuracy/loss) для обучающего множества

	Cifar10	Cifar10+Cifar100
Accuracy	0.6789	0.76542
loss	0.91403	0.6771

Очевидно, что дополнительные 10ть эпох обучения не только дообучили нейронную сеть для новых классов, но и улучшили ее показатели в целом. Добиться экстремума результатов не позволяет ограниченность временных и физических ресурсов(моих).

