

1 Обучить сверточную нейронную сеть в стиле AlexNet (с падением размера ядра свертки и последовательностью блоков свертка-пулинг (conv-pool) - (conv-pool) - ...) на датасете fashion-mnist или cifar10

Model: "sequential_22"

Layer (type)	Output Shape	Param #
conv2d_83 (Conv2D)	(None, 28, 28, 32)	320
max_pooling2d_9 (MaxPooling2D)	(None, 14, 14, 32)	0
conv2d_84 (Conv2D)	(None, 14, 14, 64)	18,496
max_pooling2d_10 (MaxPooling2D)	(None, 7, 7, 64)	0
conv2d_85 (Conv2D)	(None, 5, 5, 128)	73,856
conv2d_86 (Conv2D)	(None, 3, 3, 128)	147,584
conv2d_87 (Conv2D)	(None, 3, 3, 64)	73,792
max_pooling2d_11 (MaxPooling2D)	(None, 1, 1, 64)	0
flatten_21 (Flatten)	(None, 64)	0
dense_51 (Dense)	(None, 128)	8,320
dropout_19 (Dropout)	(None, 128)	0
dense_52 (Dense)	(None, 64)	8,256
dropout_20 (Dropout)	(None, 64)	0
dense_53 (Dense)	(None, 10)	650

Total params: 331,274 (1.26 MB)

Trainable params: 331,274 (1.26 MB)

Non-trainable params: 0 (0.00 B)

Epoch 1/6

469/469 120s 250ms/step - accuracy: 0.5348 - loss: 1.2487 - val_accuracy: 0.8276 - val_loss: 0.4517

Epoch 2/6

469/469 120s 257ms/step - accuracy: 0.8281 - loss: 0.4939 - val_accuracy: 0.8659 - val_loss: 0.3725

Epoch 3/6

469/469 138s 249ms/step - accuracy: 0.8746 - loss: 0.3840 - val_accuracy: 0.8834 - val_loss: 0.3344

Epoch 4/6

469/469 116s 247ms/step - accuracy: 0.8890 - loss: 0.3364 - val_accuracy: 0.8982 - val_loss: 0.2919

Epoch 5/6

469/469 115s 245ms/step - accuracy: 0.9028 - loss: 0.2952 - val_accuracy: 0.9051 - val_loss: 0.2728

Epoch 6/6

469/469 117s 250ms/step - accuracy: 0.9097 - loss: 0.2721 - val_accuracy: 0.9056 - val_loss: 0.2692

313/313

6s 21ms/step - accuracy: 0.9037 - loss: 0.2814

313/313

6s 20ms/step

Test accuracy 90.56%

Classification Report:

	precision	recall	f1-score	support
T-shirt/top	0.8330	0.8780	0.8549	1000
Trouser	0.9929	0.9800	0.9864	1000
Pullover	0.8351	0.8860	0.8598	1000
Dress	0.8844	0.9260	0.9047	1000
Coat	0.8665	0.8370	0.8515	1000
Sandal	0.9837	0.9660	0.9748	1000
Shirt	0.7706	0.6820	0.7236	1000
Sneaker	0.9474	0.9540	0.9507	1000
Bag	0.9858	0.9720	0.9789	1000
Ankle boot	0.9512	0.9750	0.9630	1000
accuracy			0.9056	10000
macro avg	0.9051	0.9056	0.9048	10000
weighted avg	0.9051	0.9056	0.9048	10000

2 Оценить рост точности при увеличении ширины сети (больше ядер)

№	Количество ядер	Точность нейронной сети
1	8, 16, 32, 32, 16, 32, 16, 10	73.88
2	16, 32, 64, 64, 32, 64, 32, 10	88.24
3	32, 64, 128, 128, 64, 128, 64, 10	90.56
4	64, 128, 256, 256, 128, 256, 128, 10	91.64
5	32, 64, 128, 128, 128, 128, 64, 10	90.25
6	32, 64, 64, 64, 64, 64, 64, 10	89.64
7	32, 32, 32, 32, 32, 32, 32, 10	86.42

Вариант 4 является лучшим по показателю точности, но затраты времени при обучении на 6 эпохах достигают получаса, что не очень быстро. 3 и 5 вариант являются оптимальными по скорости и точности работы нейронной сети.

№	Размер окна	Точность нейронной сети
1	2 на 2	89.40
2	3 на 3	90.56
3	4 на 4	90.53
4	5 на 5	90.92

3 Оценить рост точности при увеличении глубины сети (больше слоев)

Будет использоваться вариант 3 из предыдущего раздела.

№	Количество слоев	Точность нейронной сети
1	3 conv, 3 pool, 2 dense, 2 drop	89.73
2	4 conv, 3 pool, 2 dense, 2 drop	90.75
3	5 conv, 3 pool, 2 dense, 2 drop	90.56
4	6 conv, 3 pool, 2 dense, 2 drop	91.08

5	7 conv, 3 pool, 2 dense, 2 drop	90.43
6	6 conv, 3 pool, 3 dense, 3 drop	90.73
7	3 conv, 3 pool, 3 dense, 3 drop	89.31

4 (опционально) сравнить с точностью полносвязной сети для этой выборки

Из второго и третьего пункта был взят набор номер 3 для сверточной сети.

Model: "sequential_2"

Layer (type)	Output Shape	Param #
conv2d_9 (Conv2D)	(None, 28, 28, 32)	320
max_pooling2d_6 (MaxPooling2D)	(None, 14, 14, 32)	0
conv2d_10 (Conv2D)	(None, 14, 14, 64)	18,496
max_pooling2d_7 (MaxPooling2D)	(None, 7, 7, 64)	0
conv2d_11 (Conv2D)	(None, 5, 5, 128)	73,856
conv2d_12 (Conv2D)	(None, 5, 5, 128)	147,584
conv2d_13 (Conv2D)	(None, 5, 5, 64)	73,792
max_pooling2d_8 (MaxPooling2D)	(None, 2, 2, 64)	0
flatten_2 (Flatten)	(None, 256)	0
dense_8 (Dense)	(None, 128)	32,896
dropout_6 (Dropout)	(None, 128)	0
dense_9 (Dense)	(None, 64)	8,256
dropout_7 (Dropout)	(None, 64)	0
dense_10 (Dense)	(None, 10)	650

Total params: 355,850 (1.36 MB)
Trainable params: 355,850 (1.36 MB)
Non-trainable params: 0 (0.00 B)

Структуру полносвязной сети возьмем из предыдущих практических работ.

Model: "my_model"

Layer (type)	Output Shape	Param #
dense_161 (Dense)	(None, 128)	100,480
dense_162 (Dense)	(None, 160)	20,640
dense_163 (Dense)	(None, 150)	24,150
dense_164 (Dense)	(None, 10)	1,510

Total params: 146,780 (573.36 KB)
Trainable params: 146,780 (573.36 KB)
Non-trainable params: 0 (0.00 B)

Результаты предсказаний: Сверточная и полносвязная

Test accuracy 90.94% Classification Report:					Result acc = 87.03 Classification Report:				
	precision	recall	f1-score	support		precision	recall	f1-score	support
T-shirt/top	0.8679	0.8540	0.8609	1000	T-shirt/top	0.7697	0.8890	0.8251	1000
Trouser	0.9990	0.9630	0.9807	1000	Trouser	0.9827	0.9660	0.9743	1000
Pullover	0.8527	0.8570	0.8549	1000	Pullover	0.7768	0.8040	0.7902	1000
Dress	0.8870	0.9340	0.9099	1000	Dress	0.8671	0.8810	0.8740	1000
Coat	0.8399	0.8760	0.8576	1000	Coat	0.7773	0.8200	0.7981	1000
Sandal	0.9918	0.9710	0.9813	1000	Sandal	0.9594	0.9220	0.9403	1000
Shirt	0.7561	0.7160	0.7355	1000	Shirt	0.7411	0.5610	0.6386	1000
Sneaker	0.9640	0.9650	0.9645	1000	Sneaker	0.9087	0.9550	0.9313	1000
Bag	0.9830	0.9850	0.9840	1000	Bag	0.9714	0.9520	0.9616	1000
Ankle boot	0.9521	0.9730	0.9624	1000	Ankle boot	0.9464	0.9530	0.9497	1000
accuracy			0.9094	10000	accuracy			0.8703	10000
macro avg	0.9093	0.9094	0.9092	10000	macro avg	0.8701	0.8703	0.8683	10000
weighted avg	0.9093	0.9094	0.9092	10000	weighted avg	0.8701	0.8703	0.8683	10000

В итоге получаем 3% разницы точности. Но если посмотреть статистику по классам, сверточная нейронная сеть лучше определяет предметы чем полносвязная. К примеру, самый сложный класс для определения Shirt сверточная нейронная сеть видит с точностью 71.6% по сравнению с 56.1% у полносвязной сети. Тем самым качество определения у сверточной сети лучше.