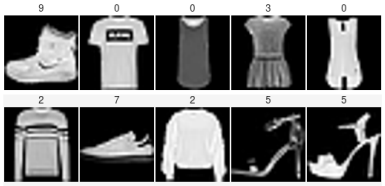
CS63 Fall 2020
Lab 6: Convolutional Neural Networks

Yael Borger
Delaney Hawkins

November 3 2022

1 Data Set

The network used a set of 60000 images, that had been reshaped into a 28 by 28 image, for training purposes, the test set included 10000 unseen images in the same size.



For the training, the network would analyze the images of the training set and classify them, compare that with the correct answers, then backpropagate to adjust what is recognized as key features that make this particular item of clothing different from the other ones. The results received from the network, after the test, include the number of parameters needed to run the network, the number of times the network did not correctly interpret the image, and the number of times the network guessed a specific item in the incorrect answers.

2 Network

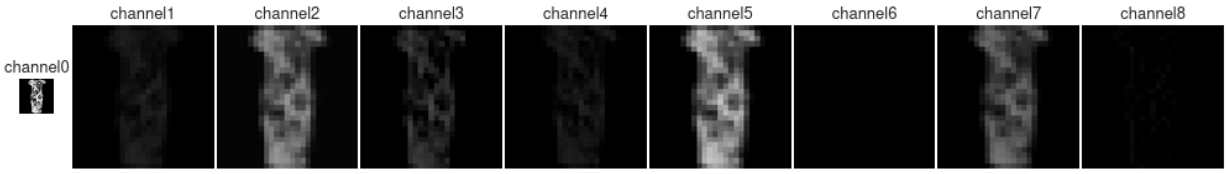
The network starts by creating a 2 Dimensional Convolutional layer, "conv1" that creates 20 kernels/layers in 2 by 2 chunks of the image and has a input shape of 28 by 28. Then we added a second Convolutional Layer, "conv2" to make 25 layers of 3 by 3 chunks. The pooling starts with "pool1," and helps to lower the number of parameters we have in the system by summarizing 2 by 2 chunks of the image. To make up for the potential loss of accuracy, a third convolutional layer, "conv3" was added, but used bigger chunks of the image, 4 by 4 chunks to be precise, and 30 layers. At that point, the network was very connected but the accuracy was not ideal, so a dropout layer was included to randomly remove some of the connections to force the network make new connections that would hopefully increase the accuracy. The concern of accuracy also led to the creation of another convolutional layer, "conv4," followed closely by another pooling "pool2", and then another convolutional layer, "conv5" to see if the accuracy would improve even after the image had been summarized in groups. The flattening was to improve the parameter amount as it had become very high at that point. The hidden layer, "hidden1," takes note of certain features of the image to highlight particular characteristics that may help narrow down the options for identification. For example, if this hidden layer specifically noticed vertical lines, it would be able to more accurately identify the difference between trousers and most of the other classification

options, since nearly all of the other options do not have a vertical gap in between other vertical shapes; the legs of the trousers are more likely to be identified as a key feature of pants. Since the image was flattened right before it, this hidden layer would likely identify these features by noting the patterned behavior, so if some item of clothing was symmetrical, the pattern would be clearer. The dropout right before the output call was part of an experimentation to see if we would lose a significant amount of accuracy from removing some links right at the end, and we did not actually lose much accuracy but did lose out on parameters. This network architecture performed the best out of the networks attempted, because it was getting the correct answers the most often, and also made the least incorrect guesses. The network was most accurate in comparison to the expected values in the tests.

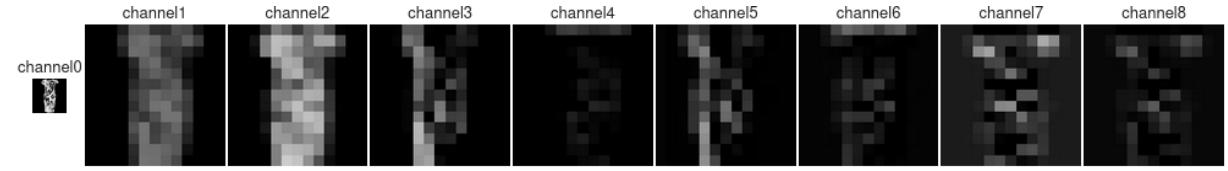
Layer (type)	Output Shape	Param #
conv1 (Conv2D)	(None, 27, 27, 20)	100
conv2 (Conv2D)	(None, 27, 27, 25)	4525
pool1 (MaxPooling2D)	(None, 13, 13, 25)	0
conv3 (Conv2D)	(None, 13, 13, 30)	12030
dropout_42 (Dropout)	(None, 13, 13, 30)	0
conv4 (Conv2D)	(None, 13, 13, 35)	26285
pool2 (MaxPooling2D)	(None, 4, 2, 35)	0
conv5 (Conv2D)	(None, 4, 2, 30)	26280
flatten (Flatten)	(None, 240)	0
hidden1 (Dense)	(None, 20)	4820
dropout_43 (Dropout)	(None, 20)	0
output (Dense)	(None, 10)	210

Total params: 74,250
Trainable params: 74,250
Non-trainable params: 0

The reason the network performed well is in part due the 2 dimensional Convolutional layers, as they often evaluated the image data using a smaller lens to look at pieces of the image separately.This allowed the network to notice features or larger patterns specific to a clothing item that can be found by looking more specifically at a chunk of pixels. The visualization of conv1 can be seen below.



Another reason it likely preformed well is its pooling layers. These acted very similarly to the convolution layers in that is helped the network to generalize information, but the pooling layers do so by averaging sections of the image together to make it store less detail. This may sound counter intuitive, but this helps the network to generalize information from one image to the next. The visualization of pool1 can be seen below.

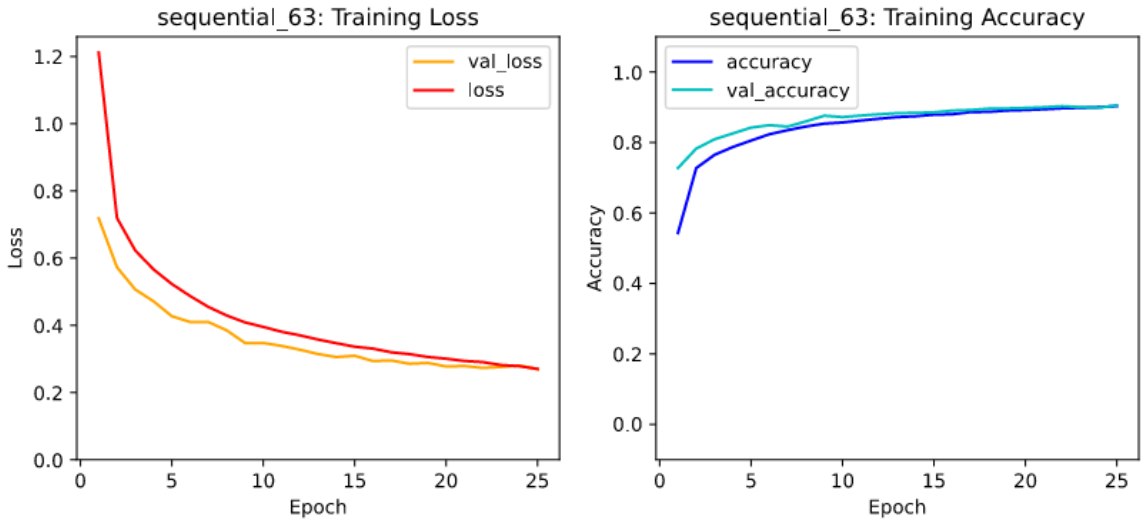


3 Training

After five runs with 25 epochs each, the networked averaged a loss of 0.2628, an accuracy of 0.9055, a val-lass of 0.2683, and a val-accuracy of 0.9048. The table below lists the results of each individual trial:

Run Number	**Loss**	**Accuracy**	**Val-Loss**	**Val-Accuracy**
1	0.2654	0.9052	0.2771	0.9029
2	0.2516	0.9101	0.2592	0.9068
3	0.2624	0.9064	0.2658	0.9051
4	0.2700	0.9036	0.2695	0.9050
5	0.2647	0.9021	0.2699	0.9044
Average	0.2628	0.9055	0.2683	0.9048

The networks rate of learning is visualized on the graphs below. The system learned more quickly at first, but had a larger difference between val-loss, and loss and val-accuracy and accuracy. However, as the learning rate slowed both losses converged to around 0.26, and both accuracy's converged to about 0.90.



4 Evaluation

Missed	Guessed
shirt(6)- missed 291 times	shirt(6)- guessed as an incorrect answer 228 times
coat(4)- missed 158 times	pullover(2)- guessed as an incorrect answer 192 times
t-shirt(0)- missed 134 times	coat(4)- guessed as an incorrect answer 181 times
pullover(2)- missed 126 times	t-shirt(0)- guessed as an incorrect answer 148 times
dress(3)- missed 96 times	dress(3)-guessed as an incorrect answer 84 times
ankle boot(9)- missed 50 times	sneaker(7)-guessed as an incorrect answer 58 times
trouser(1)- missed 33 times	ankle boot(9)-guessed as an incorrect answer 25 times
sneaker(7)- missed 30 times	sandal(5)-guessed as an incorrect answer 21 times
bag(8)- missed 20 times	bag(8)- guessed as an incorrect answer 18 times
sandal(5)- missed 19 times	trouser(1)- guessed as an incorrect answer 2

The network was particularly good at sorting through information and identifying key features of the items, such as the parts of the clothing that were distinct and took up significant portions of the image. For that reason it was able to identify the items that were clearly a different category of clothing entirely. As seen above the network did the best at identifying sandals, bags, sneakers, and trousers, all of which have a very distinct shape.

The main issue that the network faced was identifying the differences between clothing pieces that were shaped similarly. There were clothing items that were in the same shape, such as the shirt, t-shirt, pullover, and coat, but had more detail-specific differences. The network would not be able to note those details because of the Convolutional layer filters and the pooling, but it did a pretty good job of identifying the area of the body, according to the data collected. Honestly, neither of us could identify these half of the time, so it seemed fair game.