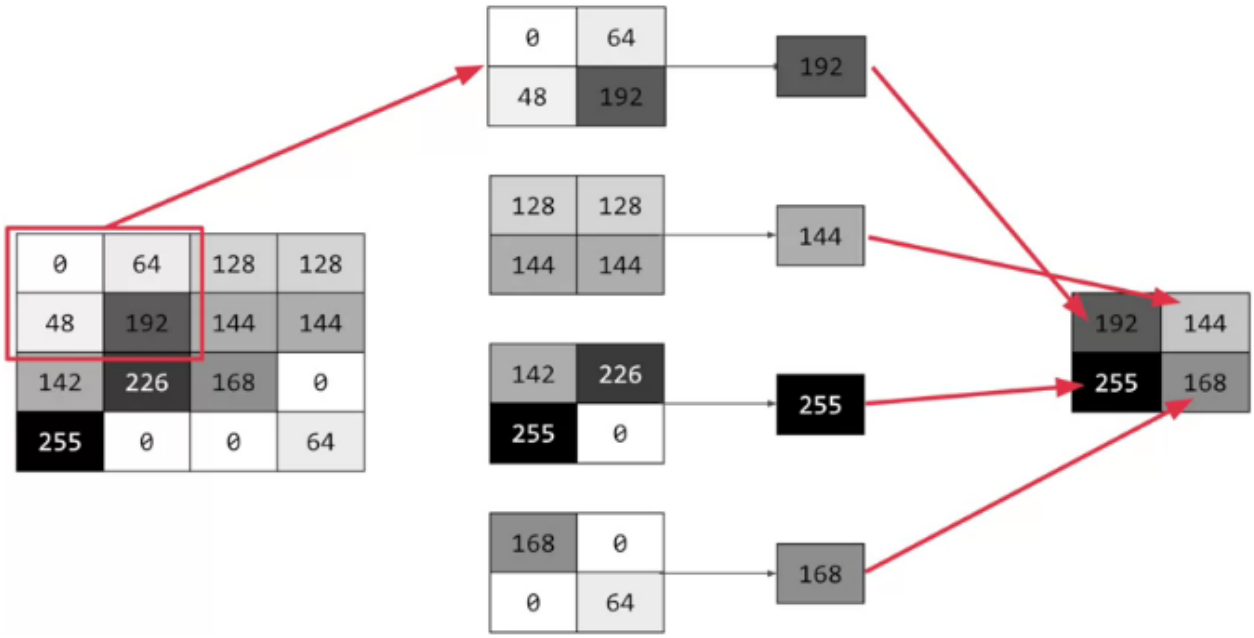
Introduction to TF for AI — Week 03 (Enhancing Vision with Convolutional Neural Networks)

aakashgoel12.medium.com/introduction-to-tf-for-ai-week-03-enhancing-vision-with-convolutional-neural-networks-7127126ab58c

February 4, 2021



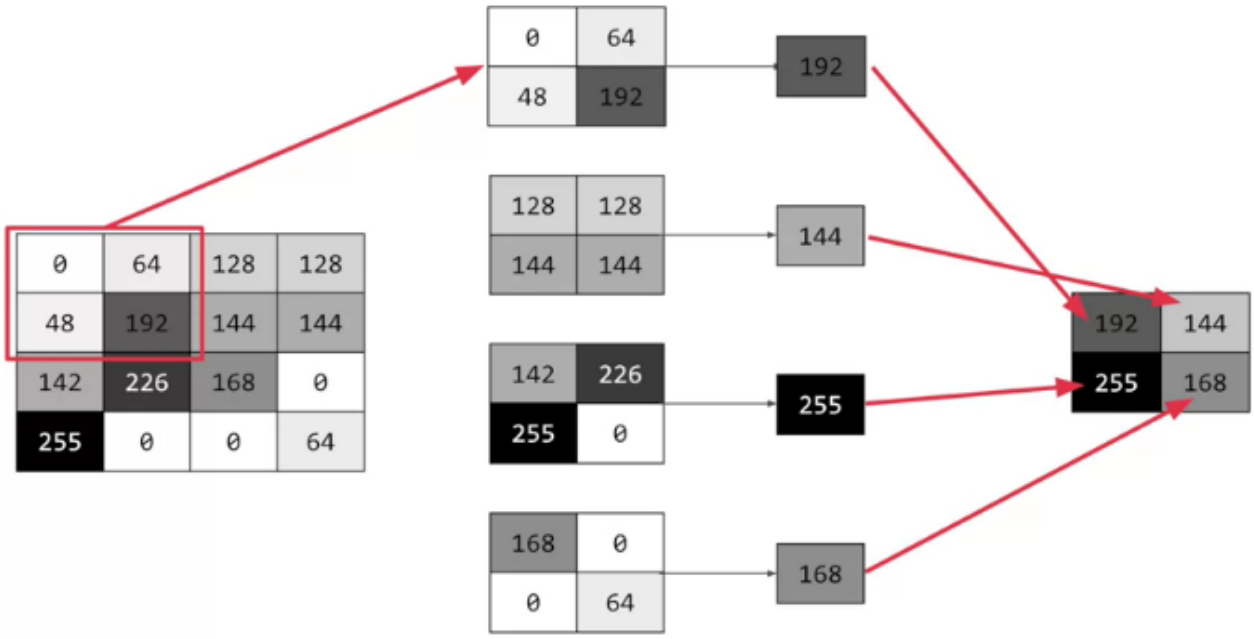
Now, one of the things that you would have seen when you looked at the images is that there's a lot of wasted space in each image. While there are only 784 pixels, it will be interesting to see if there was a way that we could condense the image down to the important features that distinguish what makes it a shoe, or a handbag, or a shirt. That's where convolutions come in.

Convolution & Pooling

Pass filter over image in order to change underlying image and multiply with

Idea here is that some convolutions will change image in such a way that certain features in image get emphasized. Like with one filter, vertical lines pop out and with other filter, horizontal lines pop out.

Pooling is one of way to compress image. Idea is to go over image of four pixels at a time and choose max out of it.




```
model = tf.keras.models.Sequential([
  tf.keras.layers.Conv2D(64, (3,3), activation='relu',
                         input_shape=(28, 28, 1)),
  tf.keras.layers.MaxPooling2D(2, 2),
  tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
  tf.keras.layers.MaxPooling2D(2,2),
  tf.keras.layers.Flatten(),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dense(10, activation='softmax')
])
```

Generate 64 filters of size 3X3, their activation is relu. In 28X28X1, 1 means we are telling that using a single byte for color depth. As our image is grey scale, using 1 byte.

MaxPooling2D(2,2) →Create Pooling Layer. For every 4 pixels, choose max out of it.

Notebook Link → https://colab.sandbox.google.com/github/lmoroney/dlaicourse/blob/master/Course%201%20-%20Part%206%20-%20Lesson%202%20-%20Notebook.ipynb

https://colab.research.google.com/github/lmoroney/dlaicourse/blob/master/Course%201%20-%20Part%206%20-%20Lesson%202%20-%20Notebook.ipynb#scrollTo=RaLX5cgI_JDb

Google Colaboratory

Edit description

QUIZ

1. What is a Convolution?

- (●) A technique to isolate features in images
- ( ) A technique to filter out unwanted images
- ( ) A technique to make images bigger
- ( ) A technique to make images smaller

✓ **Correct**

2. What is a Pooling?

- ( ) A technique to combine pictures
- ( ) A technique to isolate features in images
- (●) A technique to reduce the information in an image while maintaining features
- ( ) A technique to make images sharper

✓ **Correct**

3. How do Convolutions improve image recognition?

- ○ They make the image smaller
- ○ They make processing of images faster
- ◉ They isolate features in images
- ○ They make the image clearer

✓ **Correct**

4. After passing a 3x3 filter over a 28x28 image, how big will the output be?

- ○ 25x25
- ○ 31x31
- ◉ 26x26
- ○ 28x28

✓ **Correct**

5. After max pooling a 26x26 image with a 2x2 filter, how big will the output be?

- ○ 26x26
- ○ 28x28
- ◉ 13x13
- ○ 56x56

✓ **Correct**

6. Applying Convolutions on top of our Deep neural network will make training:

- ◉ It depends on many factors. It might make your training faster or slower, and a poorly designed Convolutional layer may even be less efficient than a plain DNN!
- ○ Faster
- ○ Slower
- ○ Stay the same

✓ **Correct**

Training

```
# GRADED FUNCTION: train_mnist_conv
def train_mnist_conv():
    # Please write your code only where you are indicated.
    # please do not remove model fitting inline comments.

    # YOUR CODE STARTS HERE

    # YOUR CODE ENDS HERE

    mnist = tf.keras.datasets.mnist
    (training_images, training_labels), (test_images, test_labels) = mnist.load_data(path=path)
    training_images=training_images.reshape(60000, 28, 28, 1)
    training_images=training_images / 255.0
    test_images = test_images.reshape(10000, 28, 28, 1)
    test_images=test_images/255.0

    model = tf.keras.models.Sequential([
            tf.keras.layers.Conv2D(32, (3,3), activation='relu', input_shape=(28, 28, 1)),
            tf.keras.layers.MaxPooling2D(2, 2),
            tf.keras.layers.Flatten(),
            tf.keras.layers.Dense(128, activation='relu'),
            tf.keras.layers.Dense(10, activation='softmax')
    ])
    class myCallback(tf.keras.callbacks.Callback):
        def on_epoch_end(self,epoch, logs = {}):
            if (logs.get('acc')>= 0.998):
                print("Reached 99.8% accuracy so cancelling training!")
                self.model.stop_training = True

    callbacks = myCallback()
    model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
    # model fitting
    history = model.fit(training_images, training_labels, epochs=20,callbacks = [callbacks])
    return history.epoch, history.history['acc'][-1]
```

Adding Convolutions to Fashion MNIST

Exploring how Convolutions and Pooling work