1. **Convolutional layer để làm gì? Lại sao mỗi layer cần nhiều kernel?**

   Lớp này được sử dụng để chiết xuất các đặc tính, chi tiết của hình ảnh đầu vào bằng các áp dụng phép toán tích chập giữa ảnh đầu vào và kernel. Sở dĩ cần nhiều kernel là vì:

   + Có nhiều đặc trưng/chi tiết của ảnh đầu vào cần phải chiết xuất, đặc biệt là nếu dữ liệu đầu vào là ảnh màu RGB (ví dụ).

   + Trong ứng dụng như nhận diện vật thể, có nhiều lớp trừu tượng. Ví dụ, để nhận diện một chiếc xe ta cần nhận diện được bánh xe và cửa xe. Và để nhận diện bánh xe, mạng neuron cần nhận diện được đường cong của bánh, tương tự với cửa xe.

2. **Hệ số của convolutional layer là gì?**

   Convolutional Layer có 3 hệ số là kích thước kernel, stride và padding.

3. **Tự tính lại số lượng parameter, output size của convolutional layer với stride và padding trong trường hợp tổng quát.**

   Kích thước input = H*W*D
   Padding = P
   Stride = S
   Kích thước kernel= F*F*D
   Số kernel = K

   - Tính size của output

     Ta xét size H của ma trận:

     - Phép Padding: size H tăng 2P
     - Áp dụng tích chập một kernel (giả sử S=1): size H giảm (F-1)
     - Giờ đây ta có $H_A = H + 2P - F + 1$
     - Phép stride: Ma trận đầu ra sẽ có size H là $[(H_A-1)/S] + 1$ với $[(H_A-1)/S]$ là kết quả số nguyên.
     - Kết luận, size H đầu ra là: ([(H+2P-F)/S] + 1)

Tương tự với size W, ta có ma trận đầu ra khi input tích chập với 1 kernel:

([(H+2P-F)/S]+1)*([(W+2P-F)/S]+1)

Với K lớp kernel thì ma trận đầu ra có kích thước:

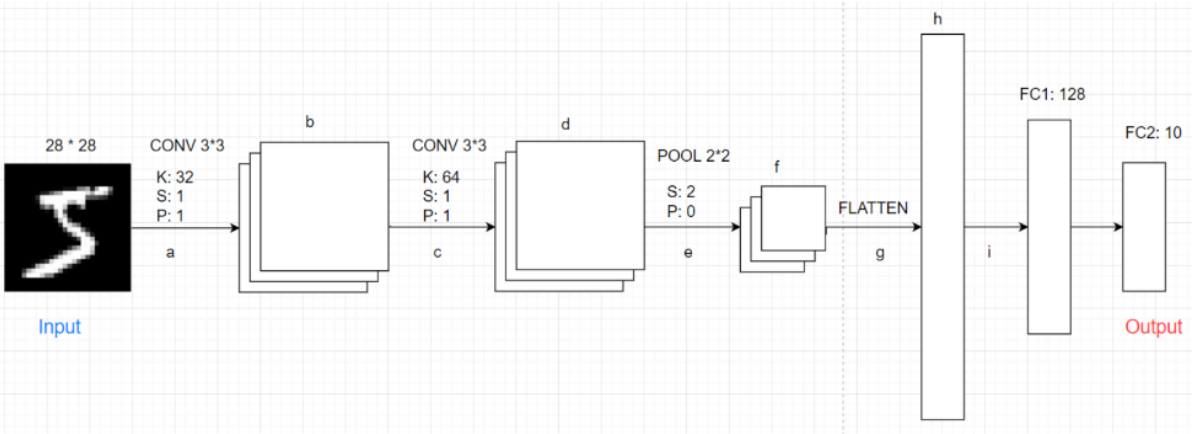
([(H+2P-F)/S]+1)*([(W+2P-F)/S]+1)*K

- Tính số lượng parameter của 1 convolution layer.
  - Số lượng parameter trong 1 kernel:

    N_1 = (F*F*D + 1) với 1 là bias.

  - Số lượng parameter trong 1 ConvLayer:
    $N = N_1*K = (F*F*D + 1)*K$

4. Hình dưới là mô hình nhận diện chữ số MNIST, K: số kernel, S: stride, P: padding. Tính số lượng parameter ở layer và output tương ứng (Tìm a, b, c, d, e, f, g, h, i).



- N_a = (3*3 + 1)*32 = 320
- O_b có kích thước (28)*(28)*(32)
- N_c = (3*3*32 + 1)*64 = 18496
- O_d có kích thước (28)*(28)*(64)
- N_e = 0 (Pooling không đóng góp vào số parameter)
- O_f có kích thước (7)*(7)*(64)
- O_f có kích thước (3136)*(1)
- N_i = (3136 + 1)*128 = 401536

5. **Tại sao cần flatten trong CNN?**

Sau khi model học được các đặc điểm trong ảnh, dữ liệu được flatten, chúng ta thực hiện bài toán phân loại bằng các fully connected layer (thực chất là các layer neuron phi tích chập). Từ đó thu về mất mát và thực hiện backpropagration.

6. **Tại sao trong model VGG16, ở layer càng sâu thì weight, height giảm nhưng depth lại tăng.**

**Trích từ:https://www.geeksforgeeks.org/vgg-16-cnn-model/*

"Thiết kế này cho phép mô hình học được các đại diện phân cấp phức tạp của các đặc tính hình ảnh, dẫn đến dự đoán linh hoạt và chuẩn xác hơn."

Em hiểu ý đồ thiết kế này theo ví dụ sau:

Để nhận dạng xe oto: