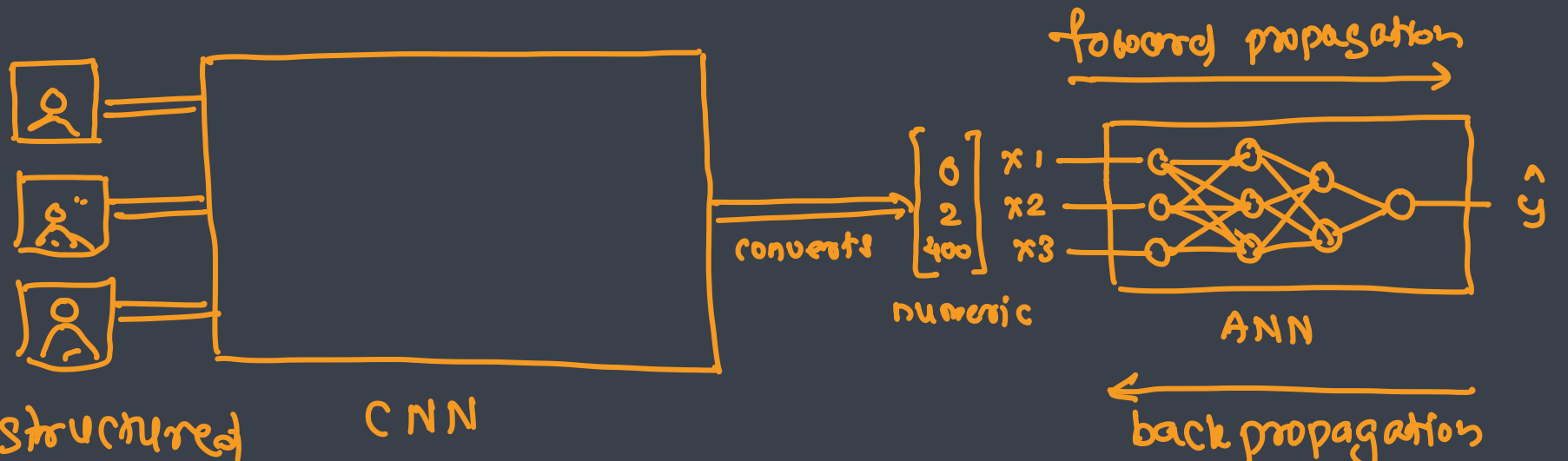


images text

CNN

image processing

text processing

History

- There have been numerous attempts to recognise pictures by machines for decades
- It is a challenge to mimic the visual recognition system of the human brain in a computer
- Human vision is the hardest to mimic and the most complex sensory, cognitive system of the brain
- In 1963, computer scientist Larry Roberts, who is also known as the father of computer vision, described the possibility of extracting 3D geometrical information from 2D perspective views of blocks in his research dissertation titled BLOCK WORLD
- In the 1970s, the first visual recognition algorithm, known as the generalized cylinder model, came from the AI lab at Stanford University
- The first visual recognition algorithm was used in a digital camera by Fujifilm in 2006

Convolutional Neural Network

- A convolutional neural network (CNN) is a type of feed-forward neural network (FNN) in which an animal's visual cortex inspires the connectivity pattern between its neurons
- In the last few years, CNNs have demonstrated superhuman performance in image search services, self-driving cars, automatic video classification, voice recognition, and natural language processing (NLP)
- In the case of real-world image data, CNNs perform better than Multi-Layer Perceptrons (MLPs)
- There are two reasons for this:
  - Unlike MLPs, CNNs understand the fact that image pixels that are closer in proximity to each other are more heavily related than pixels that are further apart:
    - CNN = Input layer + hidden layer + fully connected layer
  - CNNs differ from MLPs in the types of hidden layers that can be included in the model
  - A ConvNet arranges its neurons in three dimensions: width, height, and depth
  - Each layer transforms its 3D input volume into a 3D output volume of neurons using activation function

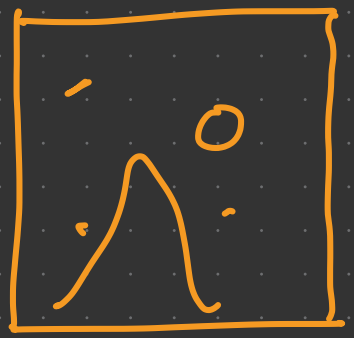
Where are CNNs used?

- CNNs are widely used in various domains, including:
  - Computer Vision: Image classification, object detection, and segmentation.
  - Medical Imaging: Analyzing medical scans like MRIs and X-rays.
  - Video Analysis: Action recognition and frame prediction.
  - Self-driving Cars: Understanding and interpreting visual data from the environment.
- CNNs have significantly advanced the field of deep learning and have become a standard approach for many visual tasks

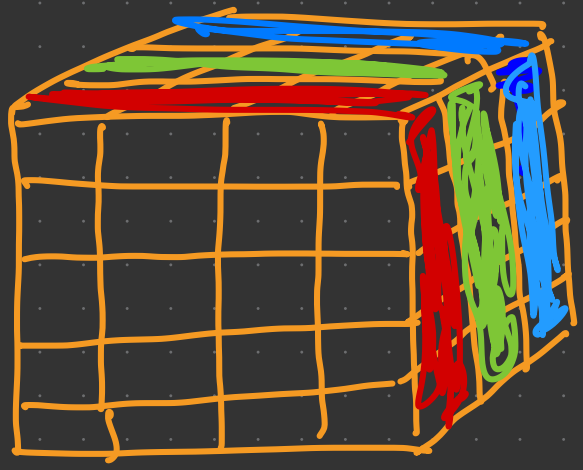
Structure of CNN

- **Convolutional Layers** → convolution → function to extract feature
  - These layers apply convolution operations to the input, using filters (kernels) to detect features such as edges, textures, and shapes. Each filter slides over the input image, producing a feature map that highlights the presence of specific features.

  information of part of image
- **Pooling Layers**
  - Pooling reduces the spatial dimensions of the feature maps, which helps to decrease computational load and control overfitting. Common types of pooling include max pooling (taking the maximum value in a region) and average pooling.
- **Activation Functions**
  - Non-linear activation functions, like ReLU (Rectified Linear Unit), are applied after convolutional and pooling layers to introduce non-linearity, allowing the network to learn complex patterns.
- **Fully Connected Layers**
  - Towards the end of the network, fully connected layers combine the features learned by the previous layers to make final predictions. Each neuron in these layers is connected to all neurons in the previous layer.
- **Dropout and Regularization**
  - Techniques like dropout may be employed to prevent overfitting by randomly dropping a percentage of neurons during training.

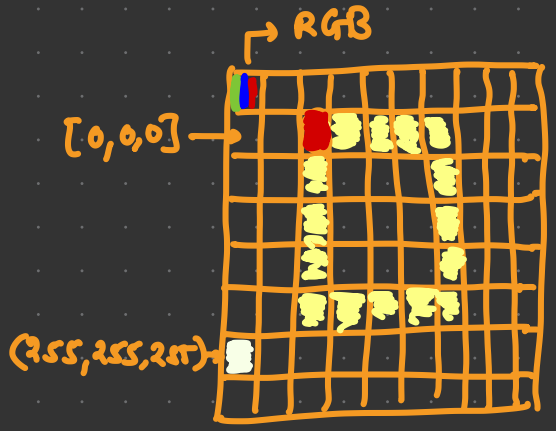


→ read →



bytes

3d array



colored

pixel = picture + Element

RGB = channels

- R = 1 byte = 2^8 = 0-255
- G = 1 byte = 2^8 = 0-255
- B = 1 byte = 2^8 = 0-255

Red = [255, 0, 0]

python → openCV, pillow

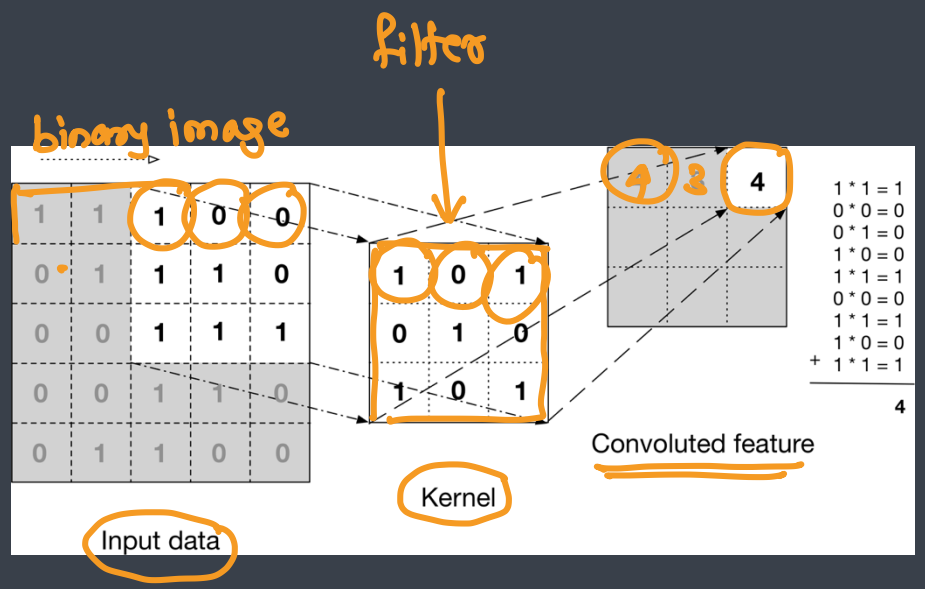
⇒

black & white

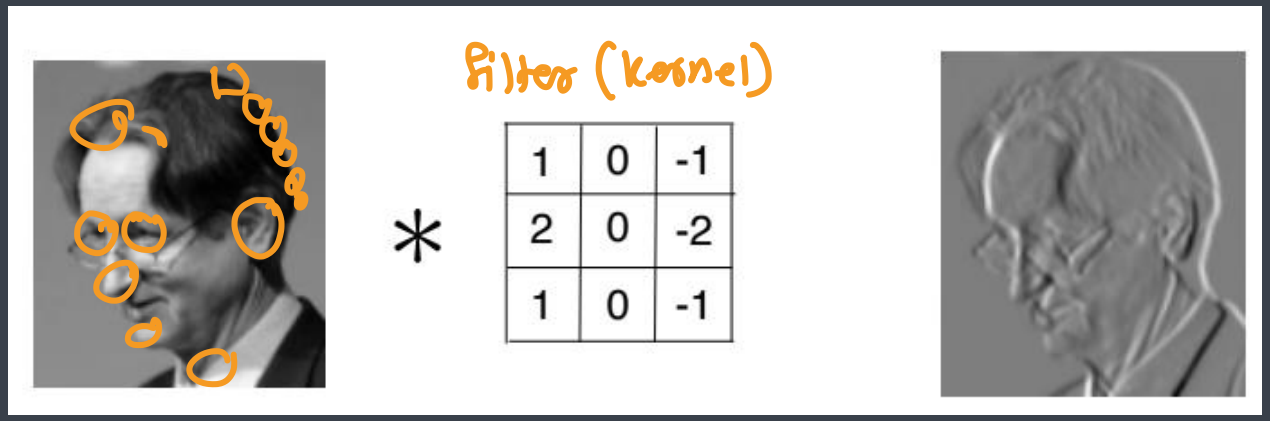
binary

Convolutional Operations

- A convolution is a mathematical operation that slides one function over another and measures the integral of their pointwise multiplication
- It has deep connections with the Fourier transformation and the Laplace transformation and is heavily used in signal processing
- The convolutional layers filter an input tensor in a tile-like fashion with a small window called a kernel
- The kernel is what defines exactly the things a convolution operation is going to filter for and will produce a strong response when it finds what it's looking for



embossing

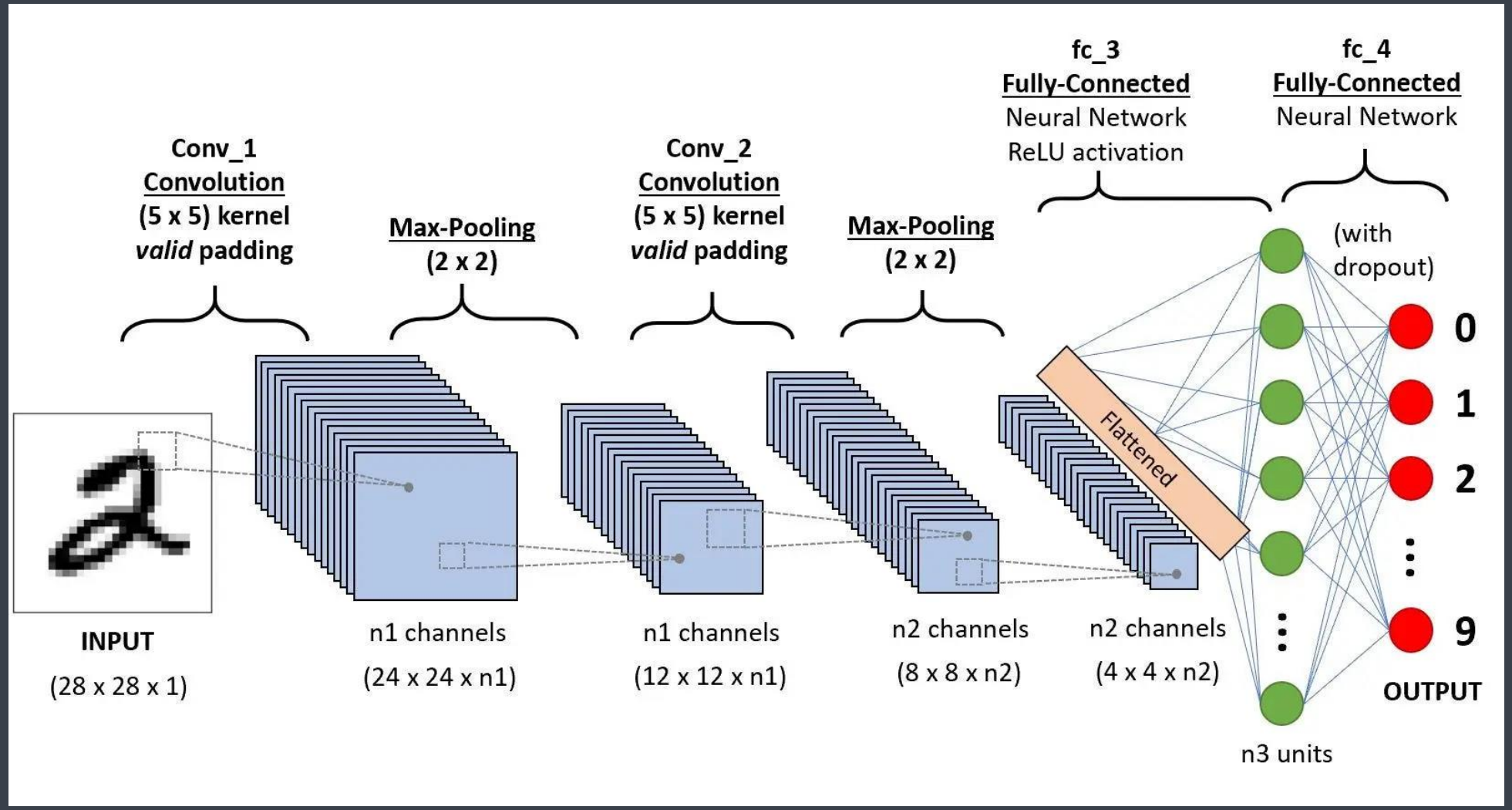


Hyperparameters of Convolutional Layer

- Kernel size (K)
  - How big your sliding windows are in pixels
  - Small is generally better, and usually odd values such as 1,3,5, or sometimes rarely 7 are used
- Stride (S)
  - How many pixels the kernel window will slide at each step of convolution
  - This is usually set to 1, so no locations are missed in an image but can be higher if we want to reduce the input size down at the same time
- Zero padding (pad)
  - The amount of zeros to put on the image border
  - Using padding allows the kernel to completely filter every location of an input image, including the edges
- Number of filters (F)
  - How many filters our convolution layer will have
  - It controls the number of patterns or features that a convolution layer will look for

```
conv1 = tf.layers.conv2d(
  inputs=input_layer,
  filters=32,
  kernel_size=[5, 5],
  padding="same",
  activation=tf.nn.relu)
```

Architecture of CNN

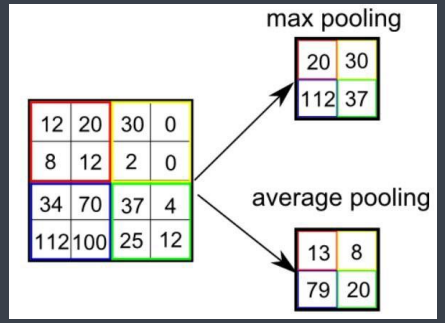


Convolution Layer

- This layer is the first layer that is used to extract the various features from the input images
- In this layer, the mathematical operation of convolution is performed between the input image and a filter of a particular size MxM
- By sliding the filter over the input image, the dot product is taken between the filter and the parts of the input image with respect to the size of the filter (MxM)
- The output is termed as the Feature map which gives us information about the image such as the corners and edges
- Later, this feature map is fed to other layers to learn several other features of the input image
- The convolution layer in CNN passes the result to the next layer once the convolution operation is applied to the input
- Convolutional layers in CNN benefit a lot as they ensure the spatial relationship between the pixels is intact

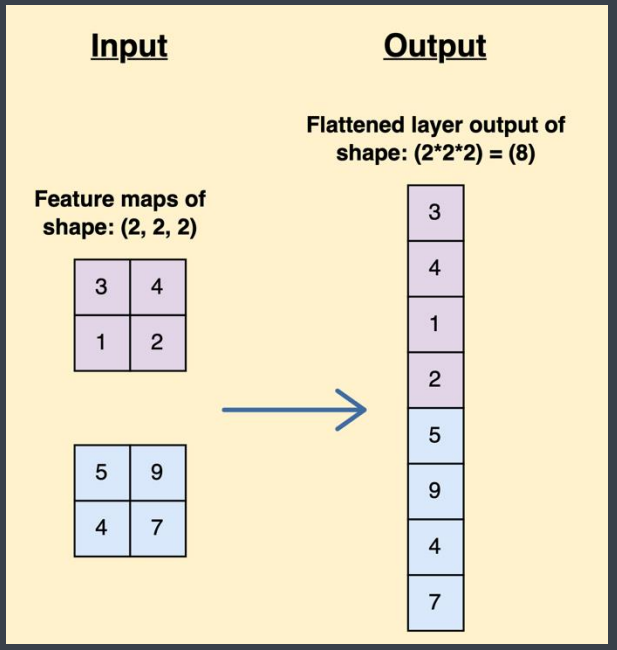
Pooling Layer

- In most cases, a Convolutional Layer is followed by a Pooling Layer
- The primary aim of this layer is to decrease the size of the convolved feature map to reduce the computational costs
- This is performed by decreasing the connections between layers and independently operates on each feature map.
- Depending upon method used, there are several types of Pooling operations
  - Max Pooling: the largest element is taken from feature map
  - Average Pooling: calculates the average of the elements in a predefined sized Image section. The total sum of the elements in the predefined section is computed in Sum Pooling
- The Pooling Layer usually serves as a bridge between the Convolutional Layer and the FC Layer
- This CNN model generalises the features extracted by the convolution layer, and helps the networks to recognise the features independently
- With the help of this, the computations are also reduced in a network



Flattening Layer

- The flatten layer is used to convert the feature map that it received from the max-pooling layer into a format that the dense layers can understand
- A feature map is essentially a multi-dimensional array that contains pixel values; the dense layers require a one-dimensional array as input for processing
- So the flatten layer is used to flatten the feature maps into a one-dimensional array for the dense layers



Fully Connected Layer

- The Fully Connected (FC) layer consists of the weights and biases along with the neurons and is used to connect the neurons between two different layers
- These layers are usually placed before the output layer and form the last few layers of a CNN Architecture
- In this, the input image from the previous layers are flattened and fed to the FC layer
- The flattened vector then undergoes few more FC layers where the mathematical functions operations usually take place
- In this stage, the classification process begins to take place
- The reason two layers are connected is that two fully connected layers will perform better than a single connected layer
- These layers in CNN reduce the human supervision

Dropout Layer

- Usually, when all the features are connected to the FC layer, it can cause overfitting in the training dataset
- Overfitting occurs when a particular model works so well on the training data causing a negative impact in the model's performance when used on a new data
- To overcome this problem, a dropout layer is utilised wherein a few neurons are dropped from the neural network during training process resulting in reduced size of the model
- On passing a dropout of 0.3, 30% of the nodes are dropped out randomly from the neural network
- Dropout results in improving the performance of a machine learning model as it prevents overfitting by making the network simpler
- It drops neurons from the neural networks during training

Activation Function

- Finally, one of the most important parameters of the CNN model is the activation function
- They are used to learn and approximate any kind of continuous and complex relationship between variables of the network
- In simple words, it decides which information of the model should fire in the forward direction and which ones should not at the end of the network
- There are several commonly used activation functions such as the ReLU, Softmax, tanH and the Sigmoid functions
- For a binary classification CNN model, sigmoid and softmax functions are preferred and for a multi-class classification, generally softmax is used
- In simple terms, activation functions in a CNN model determine whether a neuron should be activated or not
- It decides whether the input to the work is important or not to predict using mathematical operations

Popular Architectures

- There are various architectures of CNNs available which have been key in building algorithms which power and shall power AI as a whole in the foreseeable future
  - LeNet
  - AlexNet
  - VGGNet
  - GoogLeNet
  - ResNet
  - ZFNet

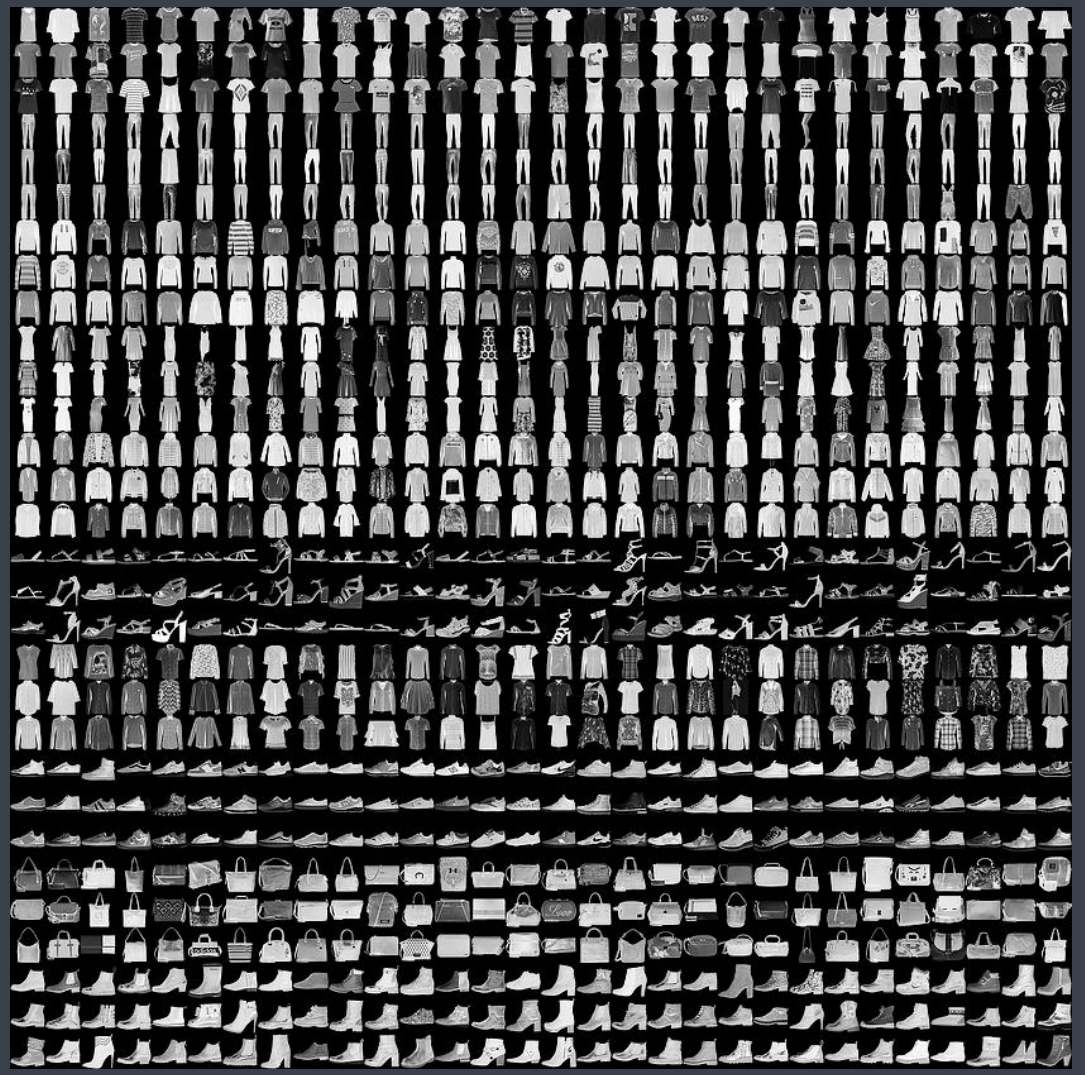
Using TensorFlow

Recognizing Clothing Items

- There are a number of different clothing items here, and you can recognize them
- You understand what is a shirt, or a coat, or a dress
- But how would you explain this to somebody who has never seen clothing? How about a shoe? There are two shoes in this image, but how would you describe that to somebody?
- This is another area where the rules-based programming can fall down
- Sometimes it's just infeasible to describe something with rules

The Data: Fashion MNIST

- One of the foundational datasets for learning and benchmarking algorithms is the Modified National Institute of Standards and Technology (MNIST) database, by Yann LeCun, Corinna Cortes, and Christopher Burges
- This dataset is comprised of images of 70,000 handwritten digits from 0 to 9
- The images are 28 × 28 grayscale.
- Fashion MNIST is designed to be a drop-in replacement for MNIST that has the same number of records, the same image dimensions, and the same number of classes—so, instead of images of the digits 0 through 9, Fashion MNIST contains images of 10 different types of clothing



Designing the Neural Network

```
# create the model
model = Sequential()

# add the flatten layer to flatten the images
model.add(Flatten(input_shape=(28, 28)))

# add the perceptron to train the model using images
# acting as an input layer
model.add(Dense(units=128, activation='relu'))

# add the output layer
model.add(Dense(units=10))

# compile the model
model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])
```