Convolutional Neural Networks

Convolutional neural networks (CNNs) are a class of **deep neural networks** that are **specialized** for **processing data** that has a **grid-like topology**, such as **images**.

- In a **fully-connected layer**, each neuron is connected to **every neuron** in the **previous layer - all activations depend on all inputs**;
- In a **convolutional layer**, each neuron is connected to **only a local region** of the **previous layer - local connectivity**;
- **Filters** are always extended along the **full depth** of the input volume **- convolve** the filter with the image (slide over the image and compute dot products);
  - Convolving a filter with an image produces an **activation map**.

Image Size, Filter Size, Stride, Channels

- **Stride**: shift in pixels between two consecutive windows;
- Number of **channels** - number of filters used in each layer;
- Given an $N \times N \times D$ image, a $F \times F \times D$ filters, and stride S, the resulting output will be of size $M \times M \times K$, where:
  - $M = \frac{N-F}{S} + 1$;
  - K is the number of filters;
- **Padding** - append zeros around the images;
  - Common padding size: $P = \frac{F-1}{2}$, which preserves the spatial size of the input $M = N$.

Convolutions

- The **convolution** of a signal x and a filter w is defined as $(x * w)$: $h[t] = (x * w)[t] = \sum_{a=-\infty}^{\infty} x[t-a]w[a]$;
- Leads to **translation/shift equivariance**;
- The second component of CNNs is **pooling** - reduces the size of the representation, which makes the network more efficient and reduces the number of parameters;
  - CNNs alternate between **convolutional layers** and **pooling layers** (provide **invariance**);

  Equivariance is a **property** of a **function** that **preserves** some **property** of the **input** in the **output**.

  Invariance is a **property** of a **function** that **does not preserve** some **property** of the **input** in the **output**.

Pooling Layer

- Makes the representations **smaller** and **more manageable**;
- **Operates** over **each activation map** independently;
- **Max pooling** - take the **maximum** value in each window;

Residual Networks (ResNets)

- **Residual networks** are a class of **neural networks** that **skip connections** - tend to lead to more stable learning;
- Key motivation: **mitigate the vanishing gradient problem**;
- With $H(x) = \mathcal{F}(x) + \lambda x$, the gradient backpropagation becomes:

$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial H}\frac{\partial H}{\partial x} = \frac{\partial L}{\partial H}\left(\frac{\partial \mathcal{F}}{\partial x} + \lambda\right)$$