

02 - Different Approaches, Layers and Architectures

Maxime Ellerbach

March 2023

End-to-End Philosophy

- Entire system is learned directly from input data without relying on hand-crafted features or intermediate representations.
- The model architecture typically consists of:
 - a feature extractor, could be a CNN as well as a regular NN (extract features from data such as images, lidar, and so on).
 - a decision-making part: uses the features extracted to make a prediction (e.g. steering, throttle, braking).
- Not reliable in some edge cases it has never seen.

- The problem is decomposed into multiple sub-problems, easier to tackle separately
- Each model solves only part of the problem. could be extracting a precise feature: lanes, road signs or handling collision detection.
- Need to interpret and filter the output data of each model to give to the next model and / or produce actions.

Lots of Layers

- Fully Connected Layers (Dense)
- Convolutional Layers (Conv)
- Recurrent layers
- Pooling layers
- Regularization

Dense layers, also known as fully connected layers, are the simplest and most common type of layer in deep learning.

- used to transform the input data by applying a linear transformation followed by a non-linear activation function
- not so great to learn to recognize feature

Convolutional Layers

- Typically used in computer vision tasks
- apply a set of filters to the input data and convolves them to generate a new set of features map. The filters are learned through the backprop
- Can be 1D / 2D / 3D - signal filtering

Recurrent Layers

Recurrent layers are used in sequence modeling tasks, such as natural language processing and speech recognition. They maintain a state vector that is updated at each time step and fed as input to the next time step. This allows the network to model dependencies between sequential data points and capture long-term context. Recurrent layers can be stacked to create deep recurrent neural networks, which can handle complex sequences with multiple levels of abstraction.

Not really used anymore in CNNs.

- Reduce spatial size of the feature maps.
- exists in multiple form: Average Pooling, Max Pooling, Global Av / Max Pooling.

Regularization Layers

Regularization layers are really useful to tackle the issue of overfitting and biases in the model

- Dropout : Randomly drop out a certain percentage of the previous layer's activation. It is used only during the training process, it is disabled while testing the neural network.
- L1 / L2: Adds a penalty term to the loss in function of the weights values. This encourages the weights to be small and with low variance, which can prevent the network from overfitting.

Lots of Architectures

- **Convolutional Neural Networks (CNNs):** Designed for processing data with a grid-like topology, such as images. Uses convolutional layers to learn local features.
- **Recurrent Neural Networks (RNNs):** Designed for processing sequential data, such as time series or natural language. Use recurrent connections to maintain state information over time.
- **Long Short-Term Memory (LSTM) Networks:** RNN that can learn long-term dependencies by selectively remembering and forgetting information.
- **Autoencoder Networks:** Encoder - compress data into a small latent ; Decoder - maps the latent representation back to original data (**non supervised**)
- **Generative Adversarial Networks (GANs):** Adversarial network to generate fake data (**non supervised**)

Convolutional Neural Network

