

Convolutional Neural Networks

Irdi Balla

11/10/2017

If your data has some structure and your model does not need to learn it from scratch then it can perform better.

1 Statistical Invariance

1.1 Translation Invariance

It does not matter where the location of the object is. A way to do that is through weight sharing. When inputs contain the same information then we can share their weight and train them jointly for those inputs.

2 Convolutional Networks (COVNETS)

COVNETS are NN that share their parameters across space. The idea of COVNETS is to take a small patch of the input and run a NN on it. The patch slides through the input without changing the weights for the NN. COVNETS are Deep Neural Networks with stacks of convolution instead of matrix multipliers. COVNETS are divided into:

- Valid padding convolutions where patch slides only inside the input.
- Same padding convolutions where you can also slide off the input and add 0 where it is not defined. When you use same padding with a stride(the shift of the patch) of 1 then the output has the same size as the input.

3 Improving COVNETS

There are different ways to improve COVNETS.

- Pooling: Instead of going aggressively with high stride we can choose a low stride and then combine the convolutions in a neighborhood. We can do so by choosing the maximal value of the neighborhood(Max-Pooling) or by averaging them(Avg-Pooling).

- 1x1 Convolution: The patch here is minimal. A 1x1 Convolution is better to be added in the middle of the usual convolution. This creates mini NN running over the patch. This is an inexpensive way to make the model deeper.
- Inception Modules: The previous models offer you the choice between one of the available options on every layer. The inception module uses all the options on 1 layer and then concatenates the output. The parameters can be chosen in a way that the total number of parameters is small but the model performs better.