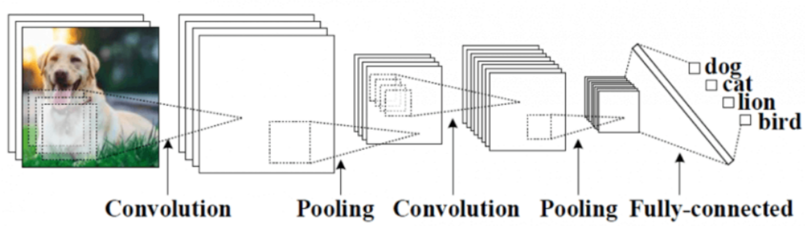
Convolutional Neural Network Implementation Issues

Convolution Neural Network (CNN) is a class of deep, feed-forward Artificial Neural Networks (ANNs) most commonly applied to analyzing visual imagery. Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. CNNs have applications in image and voice recognition, recommender systems, and natural language processing. The connections have numeric weights that are tuned during the training process, so that a properly trained network will respond correctly when presented with an image or pattern to recognize. The network consists of multiple layers of feature-detecting "neurons". Each layer has many neurons that respond to different combinations of inputs from the previous layers. The layers are built up so that the first layer detects a set of primitive patterns in the input, the second layer detects patterns of patterns, the third layer detects patterns of those patterns, and so on. Typical CNNs use 5 to 25 distinct layers of pattern recognition. Training is performed using a "labeled" dataset of inputs in a wide assortment of representative input patterns that are tagged with their intended output response. Training uses general-purpose methods to iteratively determine the weights for intermediate and final feature neurons.

The following are the implementation issues associated with CNN:

- CNNs completely lose all their internal data about the pose and the orientation of the object and they route all the information to the same neurons that may not be able to deal with this kind of information.
- A convolution is a significantly slower operation than, say maxpool, both forward and backward. If the network is pretty

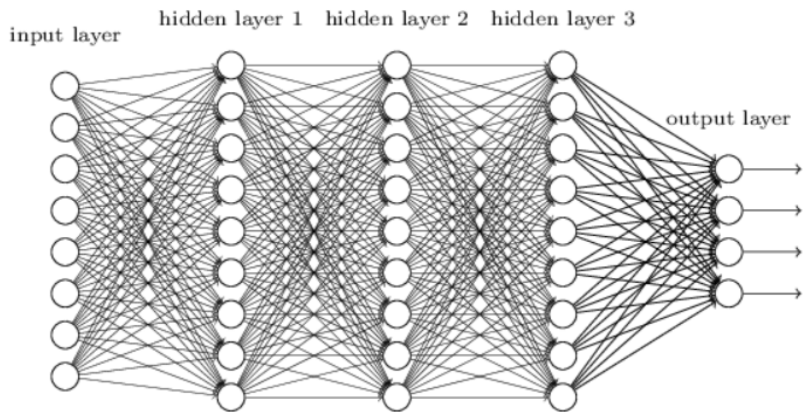
deep, each training step is going to take much longer.



The figure above depicts the structure of a CNN to predict what the animal in the image is

- From a memory and capacity standpoint the CNN is not much bigger than a regular two layered neural network. However, at runtime the convolution operations are computationally expensive and take up a lot of time (Chellapilla, Puri, Simard).
- CNNs require a large number of trainable parameters. For instance, a 24 x 24 input layer would have 600 connections per single neuron in the hidden layer.
- CNNs have little or invariance to shifting, scaling, and other forms of distortion.
- In CNNs, the topology of the input data is completely ignored. It yields similar training results for all permutations of the input vector.
- For many applications, only a small amount of training data is available. However, CNNs usually require a large amount of training data in order to avoid overfitting. A common technique is to train the network on a larger data set from a related domain.
- Once the network parameters have converged an additional training step is required to be performed using the in-domain data to fine-tune the network weights.

- If there are convoluted layers stacked with non-linearities between them, the neurons would be computing a linear function over the input, while the stacks of convoluted layers contain non-linearities that make their features more expressive.



The figure above depicts the structure of a CNN, which includes the input, hidden, and output layers

- One disadvantage of this weight sharing, in CNN, is that other pooling layers cannot be added on top of it because the filters outputs in different pooling bands are not related
- Hyper-parameter tuning is non-trivial in the CNN algorithm. It is not easy to find the best performing network structure for a specific application because it is often unclear how the network structure relates to the network accuracy (Bochinski, Senst, Sikora).
- In backpropagation problems, you might need more memory to hold the intermediate convoluted layer results.
- In CNN, to prevent overfitting of a network the training process is stopped before overfitting has had a chance to occur. It is easy

to see why this technique works, but it comes with the disadvantage that the learning process is halted.

- LeNet-5 is a pioneering 7-level convolutional network that classifies digits. But, the ability to process higher resolution images requires larger and more convolutional layers, so this technique is constrained by the limited availability of computing resources.

Chellapilla, K. Puri, S. Simard, P. High Performance Convolutional Neural Networks for Document Processing. Retrieved from: https://hal.inria.fr

Bochinski, E. Senst, T. Sikora, T. Hyper-Parameter Optimization for Convolutional Neural Network Committees based on Evolutionary Algorithms. Retrieved from: http://elvera.nue.tu-berlin.de