Deep Learning Neural Nets

Motivation

The motivation: the hidden layers are viewed as levels of abstraction. While the low levels handle the raw data, the higher levels can represent abstract high level "concepts". The argument is that useful concepts should be formed from high level abstract concepts, and not described in terms of low level data.

Example: A face can be described in terms of concepts such as eyes, mouth, roundness, etc. It seems impossible to give a direct description of a face in terms of raw pixels. Similarly eyes should not be described in terms of pixel values but in terms of other high level concepts that relate to their shape and color.

"Old" approach

Applying the error back propagation algorithm of the 80's to a network of several layers is computationally very challenging. Until about 2011 it was considered impractical, and research on deep learning was focused on alternative training methods. The main idea is as follows:

- Back propagation is too slow if its initial starting point is random. It works well if its initial starting point is near a "useful" minimum of the error function.
- In a multi-layer neural net the low level layers are expected to combine low level features to form higher level features. This can be done without knowing what the learning process is trying to achieve.
- Therefore we can start by training the network to extract "useful" features from the data. Once these features are found, use the weights values as the initial network values and train the network to identify a particular concept.
- Using this approach we are interested in techniques to train a network for "blind feature extraction".
- Hinton and his co-authors have proposed a greedy approach, based on the idea of using the network output the same as the network input. This allows extraction of a small number of features that are nearly as good as the the original features in terms of their ability to reconstruct the data.

New approach

The back propagation algorithm can be enhanced with multiple "tricks". These tricks make it run much faster and handle deep learning. The current view is still somewhat similar to the old view. Instead of starting the network training from scratch, use as a starting point another network that has already been trained on a similar problem.

Back propagation enhancements

The following is a partial list:

- ReLU as sigmoid replacement.
- Cross-entropy / softmax for evaluating error at top layer during training.
- Replace top layer with SVM for final tuneup.
- Regularization for reducing overfitting.
- Stochastic stepest-descent.
- ADAM for accelerating back propagation convergence.
- Dropouts.
- Fine tuning of random initialization.
- Convolutional layers for image input.
- Network architecture with special layers that reduce number of weights.