Discussion Quiz #0

Ashutosh Tiwari (ashutiwa@iu.edu)

January 24, 2022

1 Papers

1.1 Paper 1

Gradient-Based Learning Applied to Document Recognition

1.2 Paper 2

Understanding the Difficulty of training Deep feedforward Neural Networks.

2 Questions

2.1 Explain the sentence, "Convolutional Neural Networks (CNNs) are specifically designed to deal with the variability of 2D shapes".

"Variability of 2D shapes" means "inconsistency of 2D shapes". This inconsistency can be of numerous types, including but not limited to their size, position, color, orientation, brightness, etc. This variability can be uncontrollable(uncertain noise, distortion, transformation etc) or may very well be controllable(things placed in some orientation).

Authors discuss this to point out that CNNs tend to outperform traditional techniques because of the reason, that they can accommodate variability by design.

2.2 What are the problems/limitations in traditional pattern recognition models shown in Fig. 1 (paper 1) and how they are overcome by the method presented in this paper?

According to authors, most of traditional pattern recognition relied on systems involving two components. First **Feature Extractor**, which was responsible for generating invariant embeddings, which could later be used by **Classifier** to make predictions. Now, as authors point out, this architecture had lot of disadvantages.

1. Because of variability of input, feature extractor had to pack a lot of prior knowledge to be able to identify and filter the invariant patterns.

2. Since most of these extractor modules were designed by hand, this required a great deal of domain expertise and labor.

3. Authors were also quick to point out that accuracy of these systems largely depended on extractor module, which had to redesigned for each unique problem. An approach which was not scalable.

It is noted in paper that combination of more computational power, large datasets and advanced ML techniques are changing this. Convolutional Neural Networks remove the need of explicit feature extraction module.

2.3 All learning techniques need a minimal amount of prior knowledge about the task. How can we incorporate that prior information in the case of multi-layer neural networks?

We can incorporate prior information about the task while choosing / designing architecture of neural network. Paper talks about the different aspects of problem statements which can affect this architecture. Type of input, output format, dataset size, etc can all be considered as such driving forces.

As paper discusses, designing an architecture can include choosing type of layers and their configurations, choose activations, different modules (in case of multi module systems) etc.

2.4 What is the use of training data, validation data, and test data? Why the test data should be disjoint from training data?

Training data is used to train the learning algorithm, validation data is used to tune the hyperparameters and test data is used to evaludate the model without any bias.

Test data should be disjoint from train data as evaluation of model on training data will be biased. Test data should be a statistically identical distribution of training data which learning algorithm has not seen before, so that we can get an unbiased, relevant report of the model performance.

2.5 What is structural risk minimization? Explain how it controls the tradeoff between minimizing the training error and minimizing the expected gap between training and test error?

Paper point out that relation between train and test error can be formalized as

$$E_{test} - E_{train} = k(h/P)^{\alpha}$$

where P is training samples, h is measure of capacity of machine and α is a number between 0.5 and 1.0. Structural Risk Minimization attempts to minimize $E_{train} + \beta H(\mathbf{W})$, where $H(\mathbf{W})$ is the regularization function and β is a constant.

Since, changing $H(\mathbf{W})$ can change the capacity of accessible subsets of parameter space to the model, it helps controlling the trade-off between minimizing the E_{train} and minimizing the expected gap.

2.6 The presence of local minima in the loss function does not seem to be a major problem in practice in the multi-layer neural network, why?

Author points out that local minima in the loss function does not seem to a problem because if network is oversized for the task, the presence of extra dimensions in latent parameter space reduces the risk of unattainable regions. In simple words, it means that if number of latent dimensions is large, then it is very improbable to have a local minima, where gradient is zero in all those dimensions.

For above reason, even if there are several local minimum, there values are much much closer to each other.

Having said that it is important to note in this conversation that we should not overly rely on very large number of trainable parameters because over-parameterized neural nets can have there own problems with convergence and presence of spurious local minima.

2.7 What are the advantages of a CNN over a fully-connected architecture for handwriting recognition task as discussed in paper 1?

According to paper, following are the advantages of CNNs:

1. Since images are dense representations, they contribute a large deal of observed variables. Including latent variables (weights) from dense network in this make the total number of variables a very high number. This can be taxing for hardware infrastructure available at disposal. CNNs have very few number of latent variables as compared to fully connected networks.

2. Dense (fully connected) NNs do not have any in-variance built-in so they require training and testing data to be normalized and centered to be interpreted. Since this preprocessing is very subjective, it is difficult to get this step right. Added complexity comes from writing styles, fonts, etc.

3. Because of structure of fully connected networks, local topology and information is completely ignored. CNNs tend to conserve those local relationships.

2.8 Define the terms: (a) receptive field, (b) feature map, (c) bi-pyramid structure of a network, (d) time-delay Neural Network

(a) Receptive Field
Receptive Field of a unit in a CNN layer is the area of input which is required to produce features at a particular position in neural network.

(b) Feature Map
Set of outputs of a layer are called feature maps of that layer.

(c) Bi-pyramid structure of a network
In a CNN, in general successive layers of convolutions and sub-sampling are alternated, which results in a "bi-pyramid" structure at each convolutional layer. It looks like a bi pyramid because of convoluted feature maps in the middle acting as base and sampling on both ends of it.

(d) Time-delay Neural Network
In general NNs do not have any temporal dimensions, i.e. they are fixed in time. TDNNs are networks that have a single temporal dimension are used in classification problems which have a time dimension to them.

2.9 Explain the saturation of activation function. How can this be overcome? Can you name an activation function that does not have this problem?

Saturation of an activation function is the stagnation of weight updates. This happens when the gradient of an activation is so small that it results in almost no change in weights of a latent variable.

To completely solve this problem this problem, we can use an activation function which never reaches an asymptotic constant value, at least in as many directions possible, i.e. their gradient is never zero.

ReLU (or Rectified Linear Unit) is a good activation function which does not have this problem. It saturates only in one direction. It's form is,

$$R(z) = \max(0, z)$$

2.10 Explain gradient vanishing and gradient explosion in the context of deep neural networks?

Gradient vanishing is the situation when gradients are very low so that when they are multiplied with learning rate η they are significantly smaller and thus do no change the weights by a significant amount. This leads to a stagnation in learning and information flow in a neural network from one layer to other layer.

Gradient explosion is the exact opposite of gradient vanishing. This is a real possibility in very deep networks. In those cases gradients accumulate over several layers and then result in a very large quantity at a specific layer, which makes a network unstable. This can be solved to some extent by **gradient clipping**, where you put maximum limits on gradients' values.

References

1. The Loss Surfaces of Multilayer Networks

2. Spurious Local Minima Exist for Almost All Over-parameterized Neural Networks