Convolutional Neural Networks and Domain Adversarial Neural Networks for hand-written numbers.

Tong Shen

December 11, 2024

1 Convolutional Neural Networks (CNNs)

1.1 Overview

Convolutional Neural Networks (CNNs) are a specialized type of neural network architecture primarily designed for processing grid-like data, such as images. Unlike traditional fully connected neural networks, CNNs leverage a unique structure which can perform feature extraction, thus make it suitable for image recognition.

1.2 Architecture

The architecture of a CNN typically consists of three main types of layers:

- **Convolutional Layers:** These layers apply convolution operations using learnable filters (kernels) that slide over the input data grid. Each filter go through all the channels and detects specific features like edges, textures, or more complex patterns.
- **Pooling Layers:** Pooling layers extract the most important information inside the filter output, thus reduce the dimensions of the feature maps. Here we use max-pooling.
- **Fully Connected Layers:** These layers function to make the final prediction based on features extracted by former CNN and pooling layers.

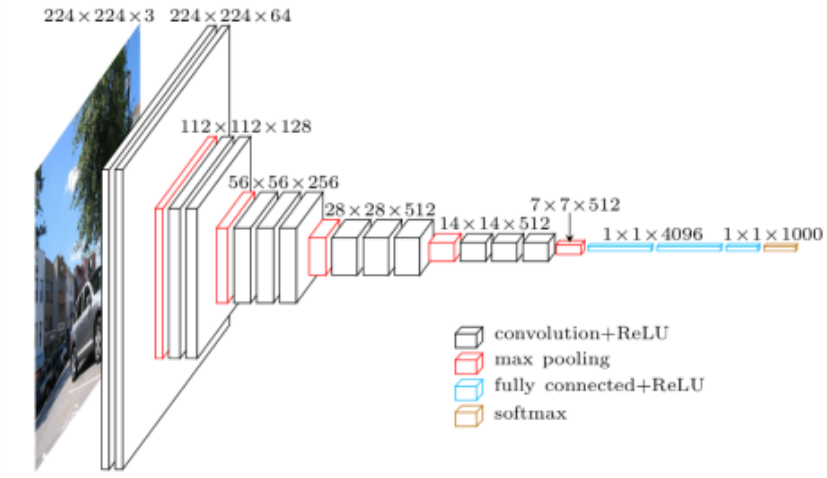

Figure 1: VGG-Net Architecture.

A typical CNN network for computer vision is a VGG-like model, whose structure is like 1. For implement, we build a VGG-like model with customized layers and size.

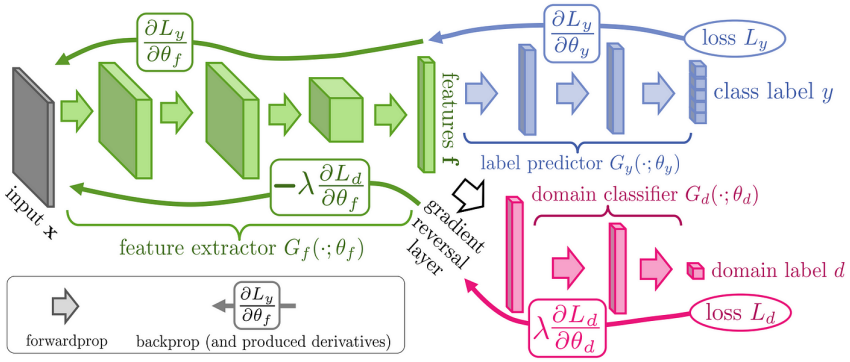

Figure 2: DANN architecture

2 Domain Adversarial Neural Networks (DANN)

2.1 Motivation

Domain Adversarial Neural Networks (DANN) address the challenge of transfer learning, where a model is trained with labelled data on one domain (source) needs to generalize effectively to a different but related domain (target) which has no or few label.

2.2 Architecture

A typical DANN architecture consists of:

- **Feature Extractor:** Learns domain-invariant features from both labeled source data and unlabeled target data.
- **Label Predictor:** Predicts the label on source data
- **Domain Classifier:** Attempts to distinguish between source and target domains

The core idea of DANN is to use a domain adversarial training strategy:

$$\min_F L_P - \lambda L_D \tag{1}$$

Where:

- F is the feature extractor
- L_P is the label predictor loss
- L_D is the domain classifier loss
- λ is a hyperparameter controlling domain adaptation strength

More specifically, the feature extractor is trying to learn the feature which both domain shares, and trying to blind the domain classifier. Meanwhile, the feature cannot be meaningless so we shoule minimize predictor loss at the same time.

A typical DANN architecture is like 2.

	CNN	DANN
MNIST	0.9891	/
MNIST-M	0.0971	0.4839

Table 1: Comparison of test accuracy

3 MNIST dataset

MNIST dataset is widely used for image recognition tasks. The original data is composed of 70,000 grayscale figures of hand-written numbers, with 60,000 for training and 10,000 for testing.

To demonstrate the effectiveness of DANN, we should made another similar dataset. We use the BSDS500 dataset to obtain random color patch background, and combine them with original mnist dataset to generate the colorful mnist-m dataset. Now we can compare the performance of our DANN model and CNN model which is only trained on grayscale figures.

4 Conclusion

The test accuracy for models and test set are list as following table 1. DANN significantly outperforms CNN whose accurancy is around a random classification accuracy 0.1. However it is still relatively low, compare to the original paper of DANN [1]. One reason is that I really don't have powerful computation resource on my laptop with RTX4060 graphic card, so hyperparameters need further adjustion.

References

[1] Yaroslav Ganin and Victor Lempitsky. "Unsupervised Domain Adaptation by Backpropagation". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, July 2015, pp. 1180–1189. URL: https://proceedings.mlr.press/v37/ganin15.html.