UNIVERSITY NAME (IN BLOCK CAPITALS)

Weather Classification

by

Junjie Zhang

A thesis submitted in partial fulfillment for the
degree of Master

in the

Chunhua Shen
School of Computer Science

October 2015

Declaration of Authorship

I, Junjie Zhang, declare that this thesis titled, 'Weather Classification' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

Abstract

Scene classification is an important field in computer vision. For similar weather condition, there are some obstacles for extracting features from outdoor images. In this paper, we present a novel approach to classify cloudy and sunny weather. Inspired by recent study of deep multi-layer neural network and spatial pyramid polling, generate a model based on imagenet dataset. Starting with parameters trained from more than 1 million images, we fine-tune the parameters. Experiment demonstrates that our classifier can achieve the state of the art accuracy.

Acknowledgements

The acknowledgements and the people to thank go here, don't forget to include your project advisor...

Contents

Declaration of Authorship i

Acknowledgements iii

List of Figures v

List of Tables vi

Abbreviations vii

Physical Constants viii

Symbols ix

1 Introduction 1
- 1.1 Overview . . . 1
- 1.2 Statistical Pattern Recognization . . . 2
- 1.3 Artificial Neural Networks . . . 2
- 1.4 Weather Classification . . . 2

2 Introduction 4
- 2.1 Single-Layer Networks . . . 4
- 2.2 Multi-Layer Networks . . . 5
- 2.3 Backpropagation . . . 7
- 2.4 Training protocols . . . 8
- 2.5 Stochastic Gradient Descent . . . 8

A Appendix Title Here 9

Bibliography 10

List of Figures

2.1 Diagram of a perceptron. 4
2.2 Diagram of a feedford neural networks. 6

List of Tables

Abbreviations

LAH **L**ist **A**bbreviations **H**ere

Physical Constants

Speed of Light $c = 2.997\ 924\ 58 \times 10^8\ \mathrm{ms}^{-\mathrm{S}}$ (exact)

Symbols

a	distance	m
P	power	W (Js^{-1})
ω	angular frequency	rads^{-1}

For/Dedicated to/To my...

Chapter 1

Introduction

1.1 Overview

Computer is one of the most significant inventions in history. It provides huge power in data processing field, like scientific computation. Also, computer system can aid human being in daily life, for example driverless vehicles. However, human brain still has compelling advantage in some fields, like identifying our keys in our pocket by feel. The complex processes of taking in raw data and taking action based on the pattern are regarded as pattern recognition. Pattern recognition has been important for people daily life for a long period and human brain has developed an advanced neural and cognitive system for such tasks.

Weather classification is one of the most important pattern recognition tasks which relates our work and lives. There are several major kinds of weather conditions, like sunny, cloudy, rainy. And people can classify them easily through eyes. However, it is not a easy job for machines, especially in computer vision literacy.

In this thesis, I describe an approach to this problem of weather classification based upon the new big trend in machine learning, and more precisely, deep learning. It differs with feature detectors method that extracts features manually and training a model to do classification.

Compared to shadow learning which includes decision trees, SVM and naive bayes, deep learning passes input data through several non-linearities functions to generate features and does classification based on the features. It generates a mapping and finds a optimist solution.

1.2 Statistical Pattern Recognization

In the statistical approach, the pattern is represented in terms of d dimensional feature vector and each element of the vector describes some subjects of the example. In brief, three components are essential to do statistical pattern recognization. First is a representation of the model. Second is an objective function used to evaluating the accuracy of the model. Third is an optimization method for learning a model with minimum errors.

1.3 Artificial Neural Networks

Artificial neural networks(ANNs) were proposed in the mid-20th century. The term is inspired by the structure of neural networks in the brain. It is one of the most successful statistical learning models used to approximate functions. Learning with ANNs yields an effective learning paradigm and has achieved cutting-edge performance in computer vision.

The single-layer perceptron has shown great success for a simple model. For increasing complex models and applications, multi-layer perceptron has exhibited the power of features learning. With hardware developing At the same time, the demanding of more efficient optimizing and model evoluation methods is needed for BIG DATA.

Recent development in ANNs approaches have giantly advanced the performance of state-of-the-art visual recognition systems. With implementing deep convolutional neural networks, it achieves top accuracy in ImageNet Challenge. The model has been used in related fields and succeeds in competition.

1.4 Weather Classification

Weather classification is an important job in daily life. In this thesis, we are focused on two class weather conditions, sunny and cloudy.

There are some obstacles in front of weather classification. First, because inputting pixels are independent and high dimension, say a 500x500 RPG image means 750000 pixels, computation is huge. Second, some simple middle level image characters are difficult to machines, like light, shading, sun shine. It is still not easy to detect these characters with high accuracy. Thirdly, there are no decisive features, for example brightness and

lightness, to classify scene. For example, sun shine can be both found in sunny and cloudy weather. Last but not least, outdoor images are various background.

Chapter 2

Introduction

2.1 Single-Layer Networks

Artificial neurons were introduced as information processing devices more than fifty years ago[**?**]. Following the early work, perceptrons were deployed in layers to do pattern recognition job. A lot of resource was invested in researching capability of learning perceptrons theoretically and experimentally. As shown in Figure 2.1, a perceptron computes a weighted summation of its n inputs and then thresholds the result to give a binary output y. We can treat n input as an vector with n elements, and represent the vector as either class A(for output +1) or class B(for output -1).

Each output is updated according to the equation:

$$y_i = f(h_i) = f\left(\sum_j w_{ij} x_j\right) \tag{2.1}$$

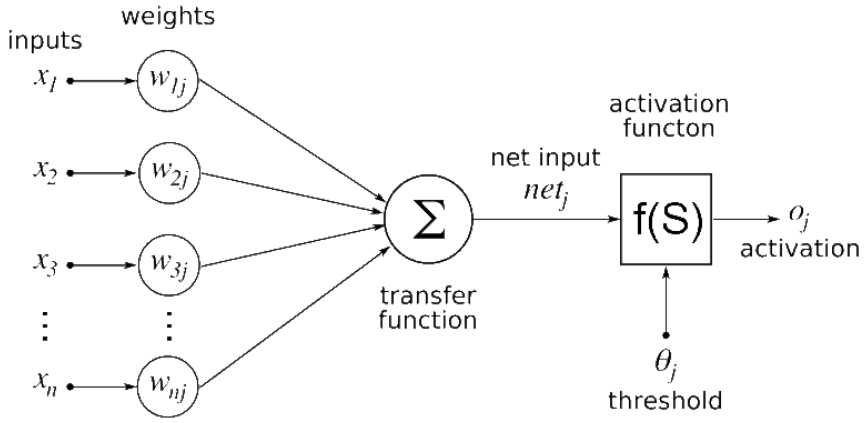

FIGURE 2.1: Diagram of a perceptron.

where y_i is the ith output, h_i is the net input into node i, The weight w_{ij} connects the ith output and the jth input, and x_j is the jth input. The function $f(h)$ is the activation function and usually makes up the form

$$f(h) = sign(h) = \begin{cases} -1 & h < 0 \\ 1 & h \geq 0 \end{cases} \tag{2.2}$$

We can also represent 2.2 in vector notation, as in

$$y = f(h) = f(\mathbf{w} \cdot \mathbf{x}) \tag{2.3}$$

where $\mathbf{w}$ and $\mathbf{x}$ can be regarded as $nx1$ dimensional column vectors, and n is the number of input data. The term $w \cdot x$ in 2.3 constructs a $(n-1)$-dimension hyperplane which passes the origin. The hyperplane can be shifted by adding an parameter to 2.1, for example

$$y = f(h) = f(\mathbf{w} \cdot \mathbf{x} + \mathbf{b}) \tag{2.4}$$

We can have the same effect by putting a constant value 1 and increase the size of x and w by one. The extra weight w with fixed value 1 is called *bias weight*; it is adaptive like other weights and provides flexibility to hyperplane. Then we get:

$$y = f(h) = f(\sum_{i=0}^{n} w_i x_i) \tag{2.5}$$

The aim of learning is to find a set of weights w_i so that:

$$y = f(\sum_{i=0}^{n} w_i x_i) = 1 \quad x \in ClassA$$
$$y = f(\sum_{i=0}^{n} w_i x_i) = 0 \quad x \in ClassB$$

2.2 Multi-Layer Networks

Single-layer networks have some important limitation in terms of the representing range of functions. We are seeking to learn the nonlinearity as the linear discriminant.To improve the representation capability, we can stack layers networks. This is the approach of multilayer neural networks. Multilayer neural networks implement linear discriminants via mapping input data to a nonlinear space. They can use fairly simple algorithms to learn form of the nonlinearity from training data.

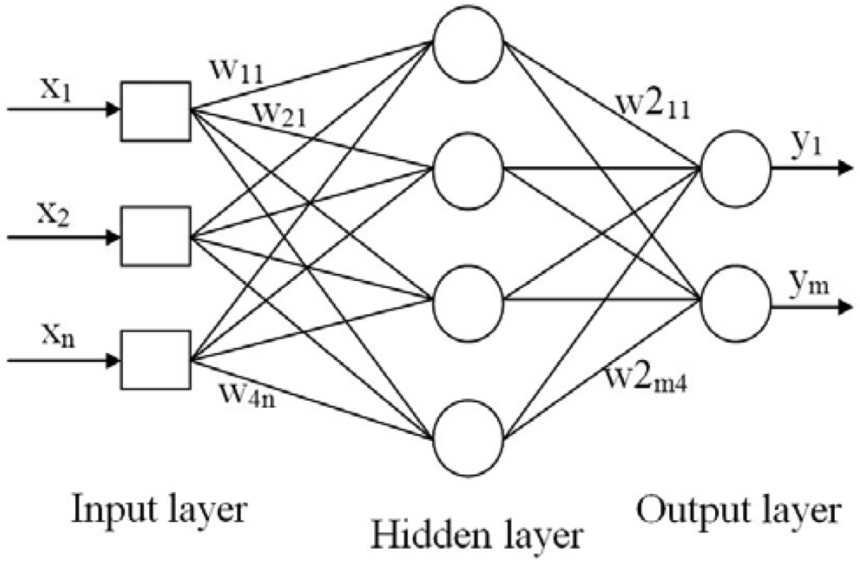

FIGURE 2.2: Diagram of a feedford neural networks.

In the thesis, we limit multi-layer networks in subset of feedforward networks. As feedforward networks can provide a general mechanism for representing nonlinear functional mapping between a set of input data and a set of labels. The 2.2 is a feedforward network having two layers of adaptive weights. In the example, the middle column perceptrons act as hidden units. The network has n inputs, 4 hidden units and m output units. The network diagram represents the function in the form

$$y_m = \hat{f}\Big(\sum_{j=0}^{m} w_{j4}^{(2)} f\big(\sum_{i=0}^{n} w_{4i}^{(1)} x_i\big)\Big) \tag{2.6}$$

In the 2.6, outer activation function could be different with the inner one.

There are some choices for activation functions, sigmoid and tanh are related and can provide high capability with continuous input data. Logistic activation function sigmoid can be represented as

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.7}$$

Its outputs lies in range $(0, 1)$. We can do a linear transformation $hatx = x/2$ on input data and a linear transformation $haty = 2y - 1$ on the output. Then we can get an equivalent activation function tanh which can be represented as

$$f(x) = tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2.8}$$

The three layer neural network is capable of approximating any function with enough

hidden units which means the networks with two layers of weights and sigmoid nonlinearities can provide any accuracy in classification problems.

2.3 Backpropagation

Multi-layer neural networks can represent mapping from input data to output classes. How to learn a suitable mapping from training dataset? And there is no explicit mapping between output and hidden units. This will be resolved by a popular learning algorithm-backpropagation.

Because networks have differentiable activation functions, the activation of output units can be propagated to hidden units with regard to weights and bias. If we have an error function, we can evaluate derivatives of the error and update the weights to minimize the error function via some optimization methods.

Backpropagation can be applied to find the derivatives of an error function related to the weights and biases in the network via two stages. First, the derivatives of the error functions, for instance sum-of-squares and Jacobian, with respect to the weights must be evaluated. Second, a variety of optimization schemes can be implemented to compute the adjustment of weights based on derivatives. After putting data forward propagation through the network, we can get the output result. It updates weight changes based on grandient descent. Suppose the network has i inputs, h hidden units and k outputs. The update equation can be represented as

$$w(j+1) = w(j) + \Delta w(j) \tag{2.9}$$

where $\Delta w(j)$ defined as

$$\Delta w(j) = -\eta \frac{\partial E}{\partial w} \tag{2.10}$$

For the hidden to output layer weights

$$\Delta w(jk) = -\eta \frac{\partial E}{\partial w_{jk}} = -\eta \delta_k y_j \tag{2.11}$$

where

$$\delta_k = \frac{\partial E}{\partial a_k} = (y_k - t_k) y_k (1 - y_k)$$

For the hidden layer weights

$$\Delta w(ij) = -\eta \frac{\partial E}{\partial w_{ij}} = -\eta \delta_j y_i \tag{2.12}$$

where

$$\delta_j = \frac{\partial E}{\partial a_j} = \sum_k \delta_k w_{jk} y_j (1 - y_j)$$

The δ_j of a hidden unit is based on the δ_ks of output units to which it links. To minimise the error function$E$ with gradient descent, it needs the propagation of errors backwards.

2.4 Training protocols

In supervised learning, we have training dataset which is data with labels. We can use the neural networks to find the output of the training data and adjust the weights to optimal values. There are mainly three types of training protocols, stochastic, batch and online training. In stochastic training, we randomly choose samples from training dataset and update weights every time depending on output of neural networks. In batch training, we use some samples and pass them through network, then update weights. In online training method, each sample of training dataset is used once and weights are updated each time. We usually call one time of passing all training dataset through neural networks one epoch.

2.5 Stochastic Gradient Descent

Because weights space in neural networks is continuous, training can reach the optimal weights value through optimization algorithms, which means minimizing loss value of function

$$L(f_w) = \sum L(y, f_w(x)) \tag{2.13}$$

Gradient descent is a method which starts from a random point, then moves to a nearby point that is downhill repeatly. It converges on a minimum possible loss.

Stochastic gradient descent is a subtype of gradient descent. It only considers a single training point and move to nearby point based on

$$w = w - \eta \Delta L(w) = w - \eta \sum_{i=1}^{n} \Delta L_i(w) \tag{2.14}$$

where η is the learning rate and $L_i(w)$ is the value of the loss function at the i^{th} sample. Although stochastic gradient descent does not guarantee convergence, it is fast.

Appendix A

Appendix Title Here

Write your Appendix content here.

Bibliography

[1] A. S. Arnold, J. S. Wilson, and M. G. Boshier. A simple extended-cavity diode laser. *Review of Scientific Instruments*, 69(3):1236–1239, March 1998.

[2] C. J. Hawthorn, K. P. Weber, and R. E. Scholten. Littrow configuration tunable external cavity diode laser with fixed direction output beam. *Review of Scientific Instruments*, 72(12):4477–4479, December 2001.

[3] C. E. Wieman and L. Hollberg. Using diode lasers for atomic physics. *Review of Scientific Instruments*, 62(1):1–20, January 1991.

[4] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013.