

# 神经网络在数据挖掘中的应用

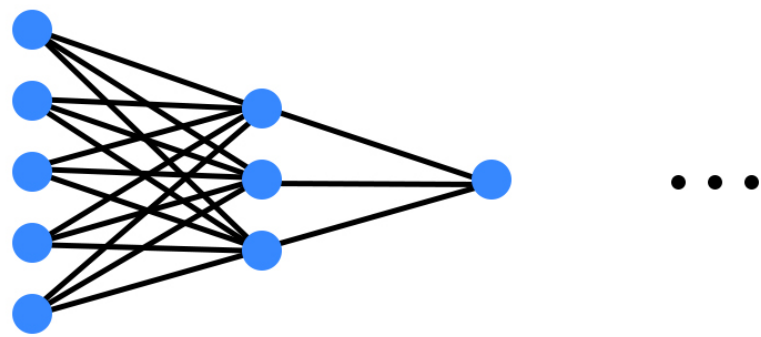
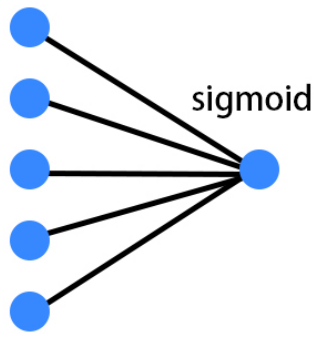
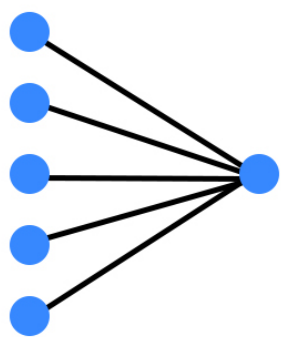
毕业论文开题答辩

数学与应用数学163班 段忠杰

# 研究内容-理论部分

## 各类神经网络的原理

1. 简单的单层神经网络（多元线性回归、对数几率回归）
2. 含隐层的神经网络（感知机）
3. 结构复杂的神经网络（递归神经网络、卷积神经网络、生成对抗网络等）



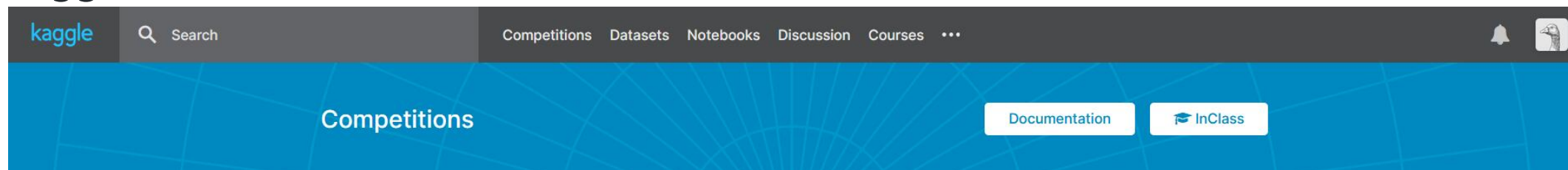
## 神经网络的相关算法

1. 梯度下降法
2. BP算法

# 研究内容-应用部分

## 数据挖掘

### 1. kaggle

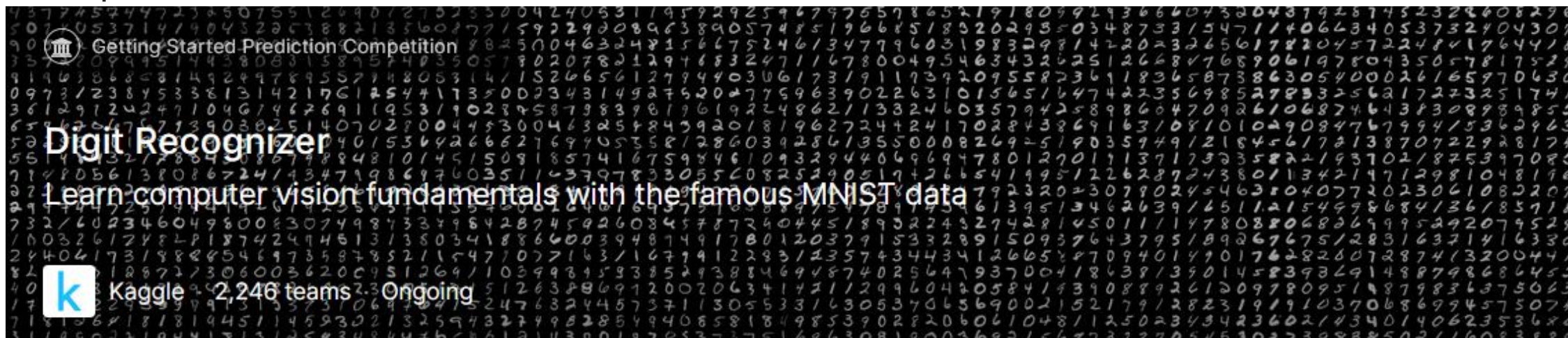



### 2. 天池



# 研究内容-应用部分

## Example

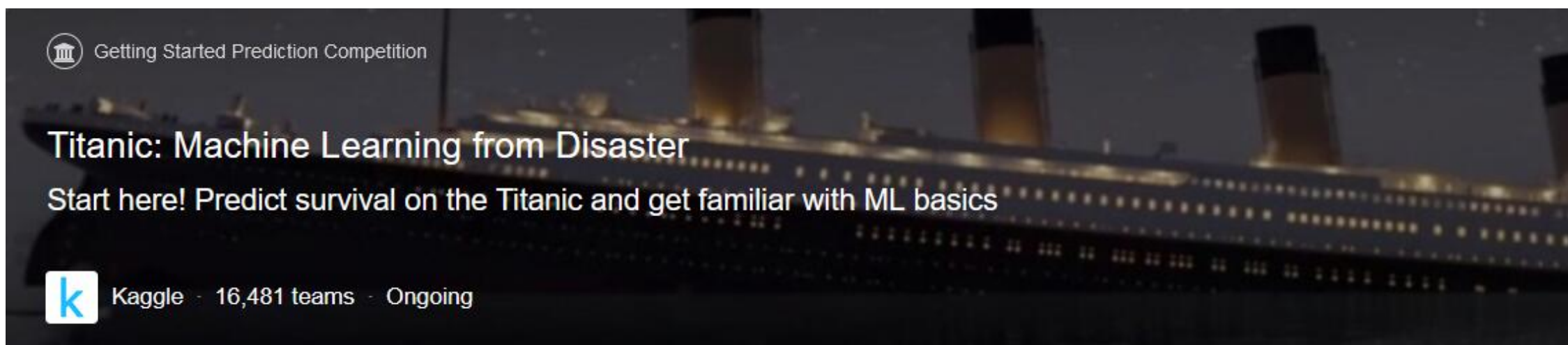



 Getting Started Prediction Competition

**Digit Recognizer**

Learn computer vision fundamentals with the famous MNIST data


 Kaggle 2,246 teams Ongoing



 Getting Started Prediction Competition













**Titanic: Machine Learning from Disaster**

Start here! Predict survival on the Titanic and get familiar with ML basics

 Kaggle 16,481 teams - Ongoing

# 研究方法

## 1. 学习机器学习相关的知识，掌握神经网络的原理

	<b>Python</b> Learn the most important language for Data Science		<b>Pandas</b> Short hands-on challenges to perfect your data man		<b>Advanced SQL</b> Take your SQL skills to the next level
	<b>Intro to Machine Learning</b> Learn the core ideas in machine learning and build your f		<b>Feature Engineering</b> The most effective way to improve your models		<b>Geospatial Analysis</b> Create interactive maps, and discover patterns in geospatial data
	<b>Intermediate Machine Learning</b> Learn to handle missing values, non-numeric values, data accurate and useful.		<b>Deep Learning</b> Use TensorFlow to take Machine Learning to the nex		<b>Microchallenges</b> Ultra-short challenges to build and test your skill
	<b>Data Visualization</b> Make great data visualizations. A great way to see the po		<b>Intro to SQL</b> Learn SQL for working with databases, using Google		<b>Machine Learning Explainability</b> Extract human understandable insights from any Machine Learning model

# 研究方法

## 2. 利用计算机编程，从原理出发实现简单的神经网络

```
61 class NeuralNetwork {
62 private:
63     Tensor dw;
64 public:
65     std::vector<NeuralNetworkLayer> ly;
66     double eta = 0.1;
67     NeuralNetwork() {}
68     NeuralNetwork(std::vector<int> siz, std::vector<int> flag) {
69         assert(siz.size() == flag.size());
70         ly.resize(siz.size());
71         ly[0].set_input_layer(siz[0]);
72         for (int i = 1; i < siz.size(); i++) {
73             ly[i].set_connect_layer(ly[i - 1], siz[i], flag[i]);
74             ly[i].weight_initialize();
75         }
76         int maxd = 0;
77         for (auto &i : siz) maxd = std::max(maxd, i);
78         dw.resize(((int)siz.size(), maxd, maxd));
79     }
80     void show()const {
81         for (int i = 0; i < ly.size(); i++) {
82             if (i == 0) {
83                 std::cout << " 0: input layer" << std::endl;
84             } else {
85                 std::cout << " " << i << ": layer" << std::endl;
86                 ly[i].show();
87             }
88         }
89     }
90     Vector predict(const Vector &x) {
91         assert(x.size() == ly[0].size());
92         ly[0].value = ly[0].activeValue = x;
93         for (int i = 1; i < ly.size(); i++) {
94             ly[i].calculate(ly[i - 1].activeValue);
95         }
96         return ly.back().activeValue;
97     }
98     void calculate_diff(int I, int J, int K) {
99         double temp;
100         for (int j = 0; j < ly[I].size(); j++) {
101             temp = active_function_diff(ly[I].activeFlag, ly[I].value[j]);
102             ly[I].diffValue[j] = (j == J) ? (ly[I - 1].value[K] * temp) : 0;
103         }
104         for (int i = I + 1; i < ly.size(); i++) {
105             for (int j = 0; j < ly[i].size(); j++) {
```

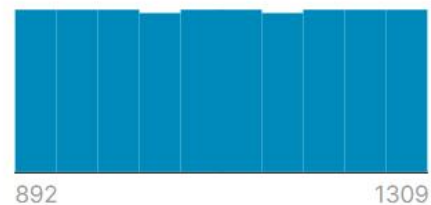


# 研究方法

## 3. 充分利用 Kaggle 等网站提供的数据，测试神经网络的性能

PassengerId

ID



Valid	418	100%
Mismatched	0	0%
Missing	0	0%
Mean	1.1k	
Std. Deviation	121	
Quantiles	892	Min
	996	25%
	1101	50%
	1205	75%
	1309	Max

Survived












Integer



Valid	418	100%
Mismatched	0	0%
Missing	0	0%
Mean	0.36	
Std. Deviation	0.48	
Quantiles	0	Min
	0	25%
	0	50%
	1	75%
	1	Max

## 研究方法

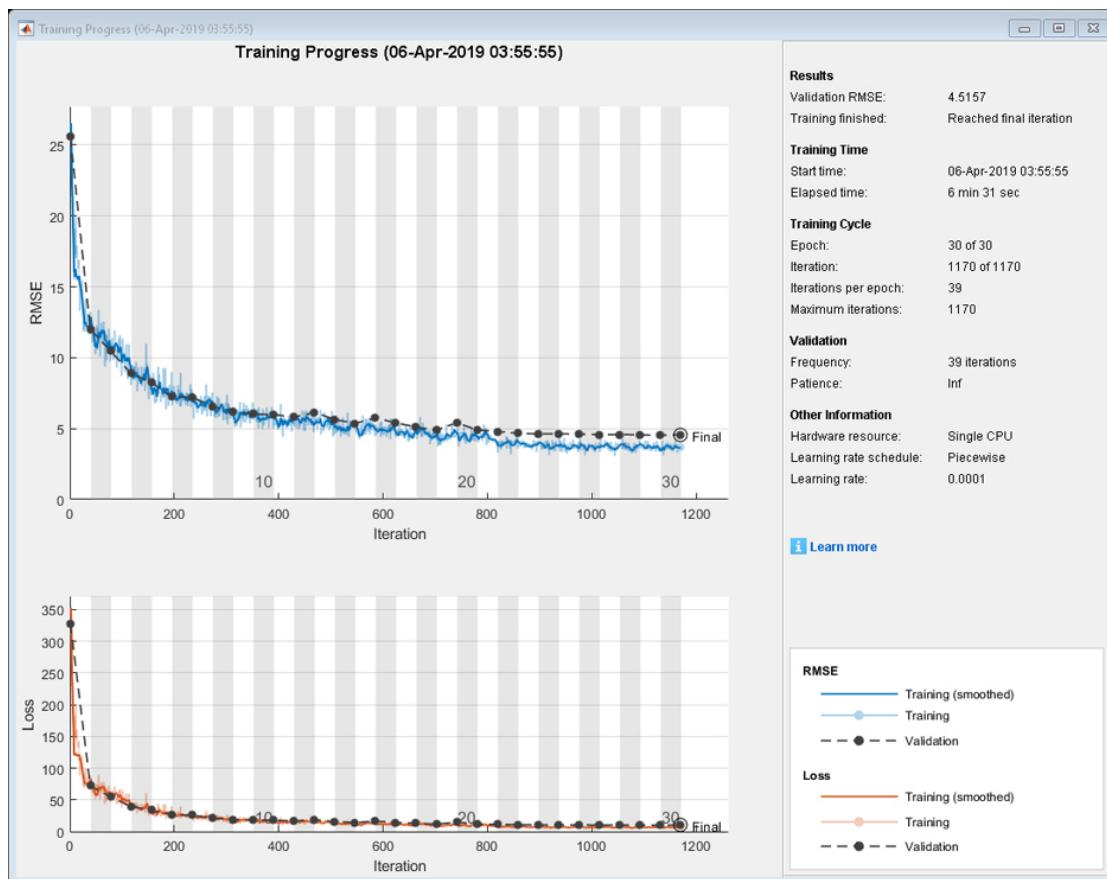
#### 4. 调整神经网络的结构与参数，提高其效率

Overview	Data	Notebooks	Discussion	Leaderboard	Rules	Team	My Submissions	Submit Predictions
1865	leiyouqi		0.95871	7	25d			
1866	Mana Masuda		0.95871	1	23d			
1867	Alexandru Zbiera		0.95871	1	14d			
1868	Alnewbie000		0.95842	4	1mo			
1869	Arnold H		0.95828	1	2mo			
1870	Talgat Omarov		0.95814	1	15h			
1871	Luke Reardon		0.95800	10	2mo			
1872	Nidhi Suthar		0.95757	2	10d			
1873	Andreas Gravrok		0.95742	2	2mo			
1874	Pan DENG		0.95671	1	2mo			
1875	dorado		0.95671	1	1mo			



# 研究方法

## 5. 使用 Matlab 与 TensorFlow 提供的神经网络框架，复现各种结构复杂的神经网络



## 端到端的开源机器学习平台

TensorFlow

针对 JavaScript

针对移动设备和 IoT 设备

针对生产

核心开源库，可以帮助您开发和训练机器学习模型。您可以通过直接在浏览器中运行 Colab 笔记本来快速开始使用。

开始使用 TensorFlow

## 研究方法神经网络

6. 尝试自行设计复杂神经网络的结构，来解决数据挖掘中的实际问题
7. 撰写论文，总结研究成果

**恳请各位老师批评指正！**