Lab 1: Backpropagation

311605010 周孫甫

1. Introduction

本次Lab1使用指定的fully-connected neural network結構對資料進行分類預測， neural network的forward pass、backpropagation皆需實作。

資料格式為一組二維資料，包含輸入、輸出，如圖所示。其中輸出以顏色表示，紅色表示數值為0的分類，藍色表示數值為1的分類。

一張含有 圖表 的圖片

自動產生的描述

Neural network結構為Fully-connected，包含2層hidden layers，以及最後的輸出層，每層neuron不定，在我的程式中，兩層hidden layer具有相同的number of unit。

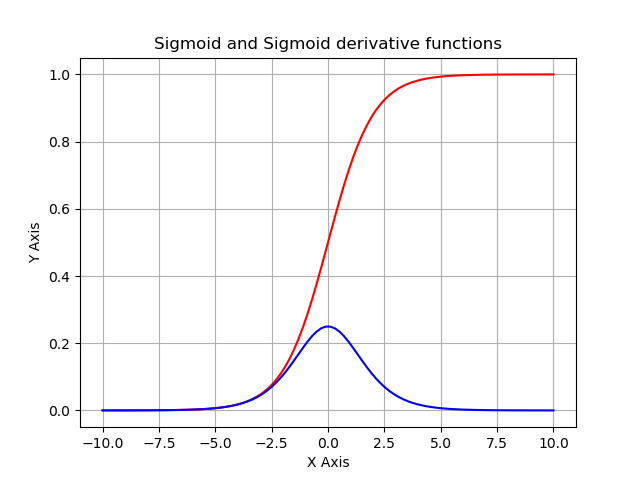
一張含有 圖表 的圖片

自動產生的描述

計算流程：

1. 產生training data
2. 決定hyperparameters，初始化parameters
3. Forward pass 得出prediction
4. Compute loss
5. Backpropagation and update parameters
6. 重複(3)~(5)直至預設epochs
7. Experiment setups
8. Sigmoid function

Sigmoid function ，



實作在actfcn.py:15

一張含有 文字, 信 的圖片

自動產生的描述

1. Neural network
2. Layers

每層Layer首先初始化時會亂數初始化所有參數並記錄activation function、optimizer，之後計算forward pass，backpropagation and update parameters都在這邊實作。

實作在nn.py:7

一張含有 文字 的圖片

自動產生的描述

1. Neural Network

Neural Network為制定各層hyperparameter，對各層依序計算forward pass，反序計算backpropagation

實作在nn.py:45

一張含有 文字 的圖片

自動產生的描述

1. Backpropagation

每層參數均被亂數初始化，之後根據loss按照backpropagation流程更新參數，實作包含在Layer中

一張含有 文字, 信 的圖片

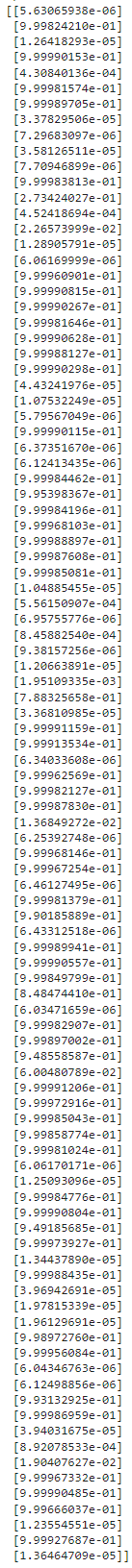
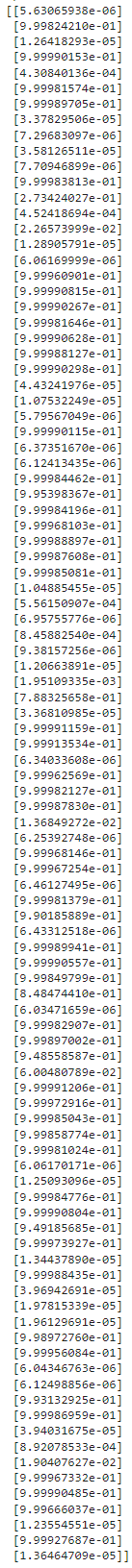
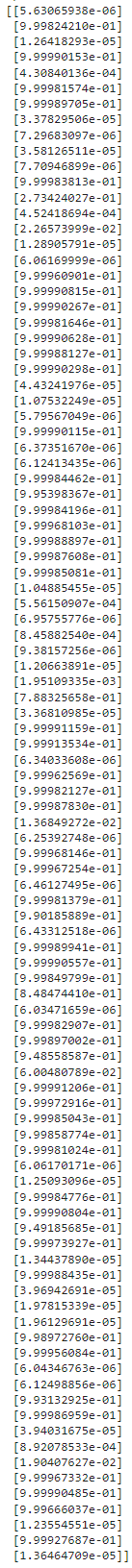
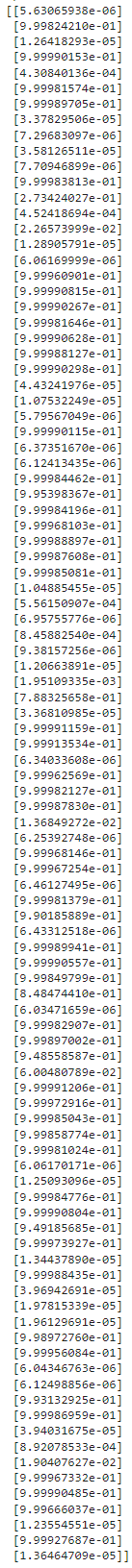
自動產生的描述

1. Results of your testing

**Linear dataset**

**一張含有 文字 的圖片

自動產生的描述**

****

**一張含有 圖表 的圖片

自動產生的描述**

**一張含有 圖表 的圖片

自動產生的描述 一張含有 圖表 的圖片

自動產生的描述**

**XOR dataset**

**一張含有 文字, 信 的圖片

自動產生的描述**

**一張含有 文字 的圖片

自動產生的描述 一張含有 圖表 的圖片

自動產生的描述**

**一張含有 圖表 的圖片

自動產生的描述 一張含有 圖表 的圖片

自動產生的描述**

1. Discussion
   * 1. **Learning rate**

實驗設計

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Dataset** | **Linear** | | | **XOR** | | |
| **Learning rate** | 0.001 | 0.01 | 0.1 | 0.001 | 0.01 | 0.1 |
| **Number of hidden units** | (4, 4) | | | (4, 4) | | |
| **Bias** | Have bias | | | Have bias | | |
| **Activation Function** | Sigmoid | | | Sigmoid | | |
| **Optimizer** | SGD | | | SGD | | |
| **Epochs** | 1.E+04 | | | 1.E+06 | 1.E+05 | 1.E+04 |
| **代號** | A | B | C | D | E | F |

|  |  |  |
| --- | --- | --- |
| 代號 | Loss | Accuracy |
| A |  |  |
| B |  |  |
| C |  |  |
| D |  |  |
| E |  |  |
| F |  |  |

觀察到兩現象，較小的learning rate會導致loss下降較慢，如A、D圖，反之則較快如C、F圖。較大的learning rate亦可能導致loss訓練時不穩定發生震盪。

* + 1. **Number of hidden units**

實驗設計

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Dataset** | **Linear** | | | **XOR** | | |
| **Learning rate** | 0.01 | | | 0.01 | | |
| **Number of hidden units** | (2, 2) | (4, 4) | (8, 8) | (2, 2) | (4, 4) | (8, 8) |
| **Bias** | Have bias | | | Have bias | | |
| **Activation Function** | Sigmoid | | | Sigmoid | | |
| **Optimizer** | SGD | | | SGD | | |
| **Epochs** | 1.E+03 | | | 1.E+04 | | |
| **代號** | A | B | C | D | E | F |

|  |  |  |
| --- | --- | --- |
| 代號 | Loss | Accuracy |
| A |  |  |
| B |  |  |
| C |  |  |
| D |  |  |
| E |  |  |
| F |  |  |

在這個部分可以看到，無論是線性資料或是XOR，適度增加neurons數量皆會導致較快收斂。

* + 1. **Without activation function**

實驗設計

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Dataset** | **Linear** | | **XOR** | |
| **Learning rate** | 0.0001 | | 0.001 | |
| **Number of hidden units** | (4, 4) | | (4, 4) | |
| **Bias** | Have bias | | Have bias | |
| **Activation Function** | Sigmoid | None | Sigmoid | None |
| **Optimizer** | SGD | | SGD | |
| **Epochs** | 1.E+05 | 1.E+02 | 1.E+05 | |
| **代號** | A | B | C | D |

|  |  |  |
| --- | --- | --- |
| 代號 | Loss | Accuracy |
| A |  |  |
| B |  |  |
| C |  |  |
| D |  |  |

對於linear dataset，不使用activation function，會有較快的收斂速度，這是因為資料簡單且線性，而不使用activation function的斜率較大能較快更新參數。

對於XOR dataset，由於是非線性資料，不使用Activation function無法處理非線性資料，因此D的loss永遠無法收斂。

* + 1. **Without bias**

實驗設計

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Dataset** | **Linear** | | **XOR** | |
| **Learning rate** | 0.01 | | 0.01 | |
| **Number of hidden units** | (4, 4) | | (4, 4) | |
| **Bias** | Have bias | No bias | Have bias | No bias |
| **Activation Function** | Sigmoid | | Sigmoid | |
| **Optimizer** | SGD | | SGD | |
| **Epochs** | 1.E+03 | | 1.E+04 | |
| **代號** | A | B | C | D |

|  |  |  |
| --- | --- | --- |
| 代號 | Loss | Accuracy |
| A |  |  |
| B |  |  |
| C |  |  |
| D |  |  |

這部份想比較的是參數僅有weight與weight+bias的比較，如A、B圖可以看到捨去bias後收斂速度反而加快，這是由於原始資料本就是簡單的線性分布，可以只簡單的使用權重表示，因此只保留比較參數加快收斂速度，但是反之XOR dataset則不然，並沒有顯著改變。

1. Extra
   * 1. Optimizers

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Dataset** | **Linear** | | | | **XOR** | | | |
| **Learning rate** | 0.01 | | | | 0.1 | | | |
| **Number of hidden units** | (4, 4) | | | | (4, 4) | | | |
| **Bias** | Have bias | | | | Have bias | | | |
| **Activation Function** | Sigmoid | | | | Sigmoid | | | |
| **Optimizer** | SGD | Momentum GD | Adagrad | Adam | SGD | Momentum GD | Adagrad | Adam |
| **Epochs** | 1.E+03 | | | | 1.E+03 | | | |
| **代號** | A | B | C | D | E | F | G | H |

Momentum:

Adagrad:

Adam:

|  |  |  |
| --- | --- | --- |
| 代號 | Loss | Accuracy |
| A |  |  |
| B |  |  |
| C |  |  |
| D |  |  |
| E |  |  |
| F |  |  |
| G |  |  |
|  |  |  |

Momentum, Adam相較SGD而言可以很快速地幫助收斂，而adagrad由於訓練後期learning rate逐漸降低，反而會導致在相同參數下的訓練速度降低，若初始learning rate較高也會有還可以的收斂表現。

* + 1. Activation functions

實驗設計

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Dataset** | **Linear** | | | | **XOR** | | | |
| **Learning rate** | 0.0001 | | | | 0.001 | | | |
| **Number of hidden units** | (4, 4) | | | | (4, 4) | | | |
| **Bias** | Have bias | | | | Have bias | | | |
| **Activation Function** | Sigmoid | ReLU | Leaky ReLU | tanh | Sigmoid | ReLU | Leaky ReLU | tanh |
| **Optimizer** | SGD | | | | SGD | | | |
| **Epochs** | 1.E+05 | | | | 1.E+05 | | | |
| **代號** | A | B | C | D | E | F | G | H |

Leaky ReLU -x方向斜率0.01

|  |  |  |
| --- | --- | --- |
| 代號 | Loss | Accuracy |
| A |  |  |
| B |  |  |
| C |  |  |
| D |  |  |
| E |  |  |
| F |  |  |
| G |  |  |
| H |  |  |

可以看到無論是ReLU、Leaky ReLU、Tanh相較sigmoid均有較快的收斂速度，在實驗中tanh的速度最快，但計算代價也最大，需要多次計算指數，ReLU計算則最快。