

Abschlussbericht Group 72 Montagsmaler

Muad, Han, Xia, John

Woche 1: In der dieser Woche haben wir das GUI gemacht. Anfangs hatten wir mehrere GUI Vorschläge, aber haben uns letztendlich für Muads GUI entschieden. Man kann mit dem GUI ein Bild zeichnen, als Images Speichern und es in einem Array umwandeln. Außerdem haben wir uns alle zu neuronale Netzwerke und deep learning informiert.

Woche 2: GUI mit Zeichenfläche, neuronales Netzwerk mit linearer Klassifizierung

-über Perzeptronen informiert, -klassifizierung backward/forward propagation entschieden -wir orientieren uns dabei an übung 2 & 3, -gui erweiterten über überarbeiten

Woche 3: in dieser Woche haben John und Xia ein Trainingsset erstellt und den Datenreader überarbeitet. Muad und Han haben Backwardpropagation und Runterskalierung gemacht.

Woche 4: In dieser woche haben wir: -weights read/write, -bias read/write, -runterskalieren
-training mit Datareader verbinden, -backwardpropagation gemacht.

Woche5: In dieser Woche haben wir die Bugs im NeuralNetwork gefixt, JavaDoc Kommentare geschrieben, das GUI mit dem NeuralNetwork verbunden, und die Präsentation erstellt.

Besonderheit und Probleme:

Problem 1: wie kann man Image im binären Code umwandeln? Früher hatten wir die Datei vom Bild im binären Code konvertiert, aber nicht das Bild im Code konvertieren.

Solution: wir konvertieren das Bild im binären Bild zuerst. Dann speichern wir jeder schwarze Pixel als 1 und weißen Pixel als 0 in einem Array.

Problem 2: mit wie viele Hidden Layer kann man das beste Ergebnis erhalten? Zuerst haben wir versucht mit 1 Hidden Layer und mit 2 Hidden Layers das Modell zu erstellen. Wir haben gefunden, dass mit 1 Hidden Layer man ein gutes Ergebnis erhalten kann. Und deswegen benutzen wir 1 Hidden Layer.

Modell

Man kann in GUI ein Bild mit Size 450*450 malen. Nach einer Bildverarbeitung kann man das Image in Datatype *Data* umwandeln. In diesem Datatype gibt es ein Array *Input[]* mit wert vom binären Pixiv und Array *Output[]* sowie String *Name* als Klassifizierung des Bildes.

In der Bildverarbeitung haben wir zuerst das Bild im binären Bild umwandelt. Dann haben wir den Bereich ohne Zeichnung entfernen und in der Größe von 50*50 skalieren. Wir haben auch geforscht, mit welcher Größe man am besten skalieren kann. Bei der Größe 50*50 gibt es am wenigsten Information Verlust. Dann konvertieren wir das Bild in 2D Array. Zum Schluss umwandeln wir in 1D Array.

Wir setzen 1D Array mit 2500 Pixels als Input Layer in Neutralem Netzwerk. Weightmatrix multipliziert mit Input Layer und addiert mit Bias Matrix erhalten wir Hidden Layer. Und analogisch erhalten Output Layer.

$$\begin{bmatrix} w_{1,1} & \dots & w_{1,2500} \\ \vdots & \ddots & \vdots \\ w_{1253,1} & \dots & w_{1253,2500} \end{bmatrix} * \begin{bmatrix} x_1 \\ \dots \\ x_{2500} \end{bmatrix} + \begin{bmatrix} b^1_1 \\ \dots \\ b^1_{1253} \end{bmatrix} = \begin{bmatrix} h^1_1 \\ \dots \\ h^1_{1253} \end{bmatrix}$$

weight matrix input bias hidden 1

Dann berechnen wir Error Matrix mit „Target – Output“. Und Berechnen wir Verteilung von Error in jedes Knoten in Hidden Layer und Input Layer. Mit $E_{hidden} = W^T + E_{output}$. Zum Berechnen Update des Weightmatrix benutzen wir $\Delta W = Lr * E * (O' * (1 - O')) \cdot HT$ mit $O' = \text{sigmoid}(O)$ und $\Delta B = Lr * E * (O' * (1 - O'))$.

Struktur

Data Class definiert ein Data Struktur. *DataReader Class* bietet die Operation von Data, und die Funktionen für Bildverarbeitung. *GUI Class* bietet die Funktionen in GUI als Viewer. *Matrix Class* bietet die mathematischen Funktionen. *Neural Network Class* bietet das Modell Neutrales Netzwerk mit training and testing Funktionen.

Wenn man das Modell training möchte. Mit *GUI* benutzt man *DataReader* und Testing Dataset in Form von *Data* lesen, dann in *Neural Network* trainieren und in Weight speichern. Wenn man *Neural Network* benutzen möchte, kann man mit *GUI* malen. Automatisch wird das Bild in *Data* umgewandelt und wird als Input von *Neural Network* gegeben. *DataReader* liest das Weight Datei ein. Nach Berechnen in *Neural Network* wird das Ergebnis von Klassifizierung in *GUI* zurückgeben.

Aufgabe Verteilung

Muad: GUI ,datareader, Neutral Network, Presentation	Han: Bildverarbeitung, Neutral Network, PPT,Wiki
John: Data, Matrix, Datareader, WeekBericht, Dataset	Weixia: Dataset, PPT, Matrix