

## Prediction

Prediction is done in predict.py

## Future roadmap for converting to latex

### Overall Architecture

- Structural Analysis: LSTM-RNN or BLSTM-RNN: 4-dim loc + label -> next loc and label expected to be scanned
- Spatial Analysis Network: ANN/SVM: take two rects and extract the spatial relationship feature
- Spatial feature to ME tree
- Tree to latex

### At this point, what we have?

Two perspective:

From the training set, we have the equation data and all of its symbols label and location.

From segmentation code, some symbols (like =) are segmented and need to be merged.

Now here just assume that the segmentation performs perfect.

### How this Architecture works

We do not know how many bounding rects are generated from the previous code, which makes it difficult to take all of those locations to directly form a math expression tree structure.

### First (B)LSTM-RNN

This RNN takes the 4-dim location vector(maybe extended to 5-dim to include the symbol label) as input. Then its output is of the same dimension: the next location to scan and expected symbol label.

The output location is a support to our structural analysis: use the output as an expectation and calculate the nearest(euclidean distance) segmented element.

The output label is a support to adjust symbol prediction of previous classifier: like x to \*, merge =, ...

## **Second Spatial Analysis Network**

From the previous RNN we got a ordered 'linkedlist' of locations data.

Each two adjacent location have a spatial relationship of the five kinds:

Superscript, Subscript, Horizontal, Vertical and Inside

Extract the 4-4 dimensions spatial relationship feature and classify it to those 5 categories.

## **Hardcoded parser**

Use the spatial relationship to construct math expression tree.

## **Tree to latex**

Yet another more trivial parser.

## **Data Preparation**

### **Extra Data needed**

- The order: 4-dim location data need to form a link
- Spatial relationship from training set need to be specified

### **User Interface to generate extra data**

- OpenCV UI: mouse down event hook
- Go back to revise choice

### **Refine on data generated from human selection**

### **Extract spatial relationship location feature**

## **Training**

- SVM/ANN to train Spatial Analysis Network
- (B)LSTM-RNN to train order
- The TWO remaining parsers: program them explicitly by human

## **Some Concerns and Other thoughts**