Checkin #1

**Handwritten Equation Identification**

Vanessa Chang (vchang5)
Divyam Dang (ddang4)
Devansh Gupta (dgupta18)

**Introduction**

The problem that is trying to be solved is classifying and identifying mathematical symbols from equations. The paper being implemented is _Recognition of Online Handwritten Mathematical Expressions Using Convolutional Neural Networks_ http://cs231n.stanford.edu/reports/2015/pdfs/mohan_lu_cs231n-project-final.pdf by Catherine Lu and Karanveer Mohan. The objective of this paper is to provide a faster alternative for recognizing math symbols and equations instead of using typesetting systems. This paper was chosen because of the applicable purpose of the results. Having experienced the long hours necessary using typesetting systems and editors, developing a way that would speed up the process seemed like a worthwhile endeavor. This problem is a classification problem because the model is trying to identify the different math symbols in an handwritten equation and present

**Related Work:**

This research paper
https://www.researchgate.net/publication/331617158_Recognition_and_Solution_for_Handwritten_Equation_Using_Convolutional_Neural_Network
could help us achieve our extended/reach goal of solving complex mathematical equations. This research paper not only addresses the identifying mathematical symbols, but it also implements a method to solve these equations, specifically quadratic equations.

***List of Useful Links***

Uses an SVM: https://github.com/anushreedas/Handwritten_Math_Expression_Recognition

Only Data Extraction: https://github.com/ThomasLech/CROHME_extractor

**Data:**
Link to dataset :
http://www.iapr-tc11.org/mediawiki/index.php/CROHME:_Competition_on_Recognition_of_Online_Handwritten_Mathematical_Expressions

The CROHME dataset consists of inkml files representing different mathematical equations consisting of over 100 different mathematical symbols.
The size of the whole CROHME file is 55mb consisting of 3 different datasets from 3 different years. This data will require some preprocessing as all the images are .inkml files so they need to be converted to a form that can be used for testing/training.
For preprocessing there are 3 main steps:
1. Data enrichment - Add distorted version of the images into the dataset
2. Image Segmentation - Which traces of the .inkml file comprise of a full symbol
3. Data Extraction - We change the InkML data comprising individual mathematical symbols into normalized images represented as pixel arrays. Then, we blur the pixels and include the number of strokes as an additional feature.

**Methodology**:

We will be training our model using CNNs. Mimicking the paper linked above, we will be tuning a 2-layer fully connected neural network. The architecture for the CNNs suggested in the paper is of the form [Conv-Relu-Pool] x N - [FC - Relu] x M - [FC] - [Softmax], where the parameters for our values of N
and M are filter size (F) and number of hidden neurons per layer (H). The number of filters used is suggested to be 32 with a 2x2 max pool with stride
of 2 for the pooling step.

Hardest part about implementing the model:
Since we are not familiar with the inkML format of the files, the hardest part would be Data extraction. Data Extraction would require us to first blur the pixels and include the number of strokes as an additional feature. All the images would not have the same dimensions and the pixel arrays would have to be normalized to a size of 24x24 pixels for the classifier.

**Metrics**:

*What constitutes "success?":*

A success constitutes when the accuracy is over a certain threshold.

*What experiments do you plan to run?*

We would experiment over the hidden neurons of the neural network and use a varying number of hidden neurons to get an accuracy closest to the model of the paper.

We could switch the pixel array size from 24x24 pixels to 32x32 pixels.
For character-classification, we could include CNN error analysis and for expression-level classification, we plan to experiment by including the hidden Markov models, as depicted in the paper listed above.

**Accuracy:**

For our project the “accuracy” of the classification of mathematical symbols is relevant. Would probably compare the number of correct symbols over the total number of symbols

**Previous Quantification:**

The authors of the research paper we have taken inspiration from aimed to change handwritten expressions into LATEX has applications for consumers and academics. They compared their results to other systems by analyzing train and test accuracies of their models on similar or same datasets.
If you are doing something new, explain how you will assess your model’s performance. Essentially we will be training our model using CNNs to produce a high accuracy in classifying mathematical symbols correctly. Hence, we will evaluate our project on the basic testing and training accuracies.

**Base, target, and stretch goals:**

Our base goal is to create a model which has a “respectable” accuracy in correctly identifying mathematical symbols present in mathematical equations. Our target is to optimize our model to get a high accuracy by experimentation. Our stretch goal may be implementing a solver that allows us to identify complex equations and provide us with an output after computation.

**Ethics:**
*Why is Deep Learning a good approach to this problem?*

Deep Learning is a good approach to this problem because it involves recognition and classification. CNNs are the optimal method when it comes to training a model to recognize and learn the data that is being inputted. CNNs are also more efficient and accurate in producing results when the problem involves classifications.

*How are you planning to quantify or measure error or success? What implications does your quantification have?*

The method for quantifying success is to identify the number of correct symbols classified and divide that over the number of total symbols. A success would be if the accuracy of the model was over a certain threshold. There should be no major implications with this method of quantification because it simply takes into account the number of correct symbols the model identified over the total numbers of symbols in the dataset.

**Division of Labor:**

Vanessa: write up, preprocess, model, train/test

Divyam: write up, preprocess, model, train/test

Devansh: write up, preprocess, model, train/test