English Character Recognition

Computer Science and Engineering

Artificial Intelligence

Artificial intelligence (AI) is the simulation of human **intelligence** processes by machines, especially computer systems. These processes include learning (the acquisition of information and rules for using the information), reasoning (using rules to reach approximate or definite conclusions) and self-correction.

Artificial neural networks (ANN) are computing systems vaguely inspired by the biological neural networks that constitute animal brains.

Machine learning

Machine learning is the scientific study of algorithms and statistical models that computer systems use to effectively perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence.

“Machine Learning is said to learn from experience E w.r.t some class of task T and a performance measure P if learners performance at the task in the class as measured by P improves with experiences.”

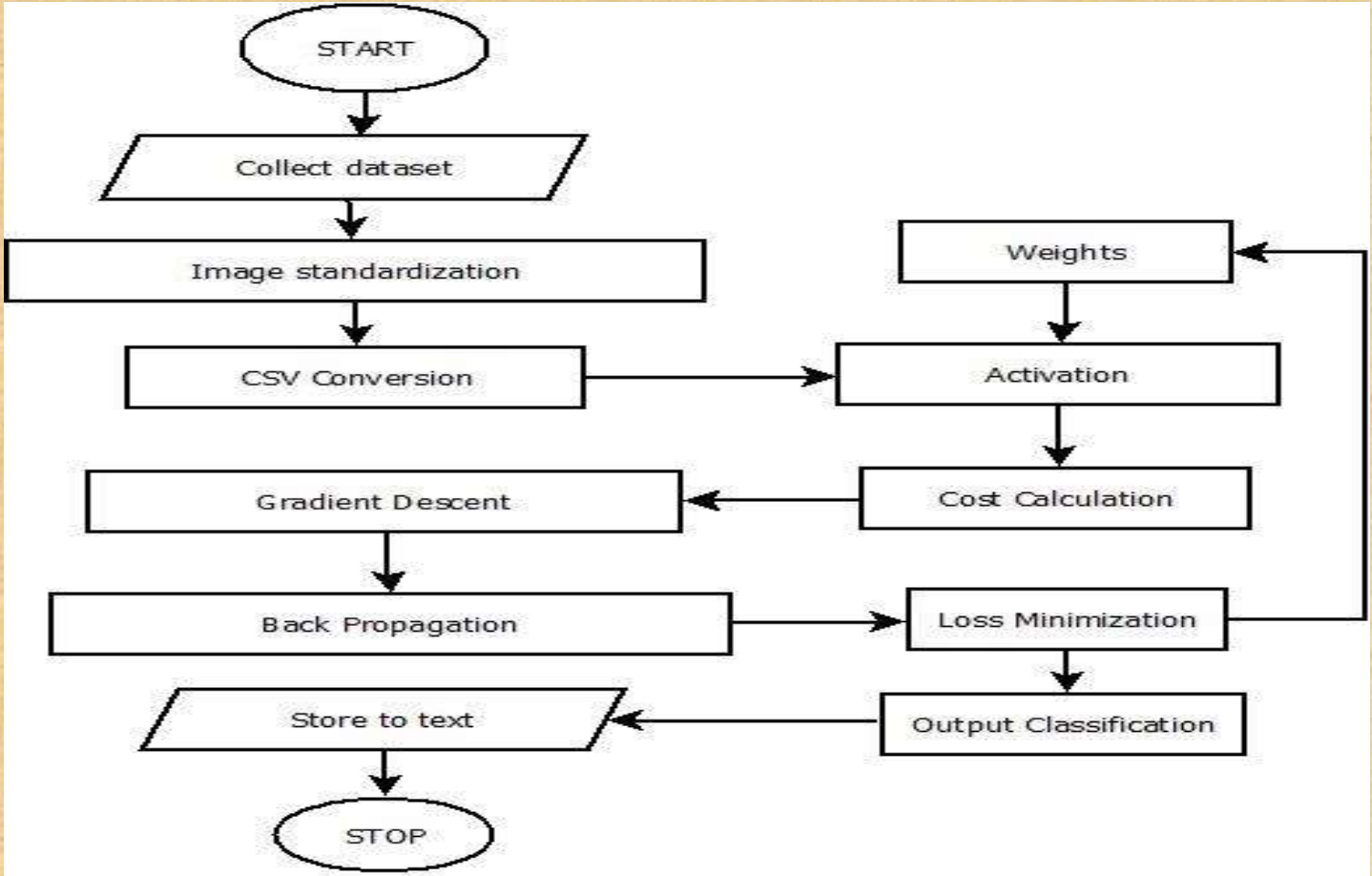
Deep Learning

Deep learning is an aspect of artificial intelligence (AI) that is concerned with emulating the learning approach that human beings use to gain certain types of knowledge. At its simplest, deep learning can be thought of as a way to automate predictive analytics.

While traditional machine learning algorithms are linear, deep learning algorithms are stacked in a hierarchy of increasing complexity and abstraction.


HELLO
WORLD

Flowchart


START
Collect dataset
Image standardization
CSV Conversion
Weights
Activation
Cost Calculation
Gradient Descent
Back Propagation
Loss Minimization
Output Classification
Store to text
STOP

Training Dataset

We are using dataset that is the form of csv taken from kaggle.com.

The dataset contains the 372450 image of 28*28 pixels converted into csv form. The data contains 26 classes of the data that are the 26 capital letters of the English alphabet.

Classifying such a big dataset into so many classes requires the neural network to be big and well trained. To accomplish this we have used deep learning consisting of 3 hidden layers where the 1st layer contains 100 neurons and the 2nd layer contains 50 neurons and the last layer of the output layer contains 26 neurons for the 26 classes. Where the input layer has 784(28*28) units.

Test dataset

For test dataset we have taken images of the letters and converted that image into 28*28 pixels and then into csv form for our model. To do that we used the following code:

```
import cv2
import numpy as np
from PIL import Image
from resizeimage import resizeimage
with open('location of the target image', 'r+b') as f:
    with Image.open(f) as image:
        cover = resizeimage.resize_cover(image, [28, 28], validate=False)
        cover.save('location of the new (changed) image', image.format)
img = cv2.imread('location of the new (changed) image')
grayscaled = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
npArray = np.array([ elem for singleList in grayscaled for elem in singleList])
np.savetxt('location of the CSV file', npArray, delimiter=",")
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Tensorflow

Google's Tensorflow is the most famous deep learning library in the world. It is called Tensorflow because it takes input as a multi-dimensional array, also known as **tensors**. You can construct a sort of **flowchart** of operations (called a Graph) that you want to perform on that input. The input goes in at one end, and then it flows through this system of multiple operations and comes out the other end as output. It allows developers to create large-scale neural networks with many layers.**TensorFlow** is mainly **used** for: Classification, Perception, Understanding, Discovering, Prediction and Creation.

The model

- The weights and biases are initialized with the Xavier Initializer and later trained.
- The forward propagation has layer with the following activation function sequence: Linear-> Relu -> Linear -> Relu -> Linear -> Softmax.
- To compute the cost we have used softmax cross entropy.
- For backpropagation we have used Adam Optimizer with learning rate 0.0001.
- The training data has been divided into mini-batches of size 32 to provide the model with extra clarity of the data. To create the mini-batches the dataset is first shuffled and then portioned.

Accuracy

The model works very well for the training data with the accuracy of 99.457%.

While the model works very great with the training data it does not work so well with the test data.

The reasons for the model not working well with the test data are:

- The test data is very small as compared to the training data.
- The model uses only deep learning and not the convolutional Neural network.
- The train data and the test data are from different dataset.
- The model is overfit for the train data so doesn't work well with the test data.
- The model doesn't use dropout and regularization to improve the model (Which we intend to use but for the lack of time we are sticking to the model we have right now).

Prediction

To predict the image provided we have developed a function that takes the trained parameter and the test image as input and predicts the class of the image.

Before the image is passed into the prediction function it goes through the resizing and csv conversion with is done right at the moment of the testing using the following code:

```
import scipy
from PIL import Image
from scipy import ndimage

my_image = "k.png"

#Image preprocessing to fit the algorithm.
fname = "/content/gdrive/My Drive/Font/" + my_image
image = np.array(ndimage.imread(fname, flatten=True))
my_image = scipy.misc.imresize(image, size=(28,28)).reshape((1, 784)).T
my_image_prediction = predict(my_image, parameters)

plt.imshow(image)
print("Your algorithm predicts: y = " + str(np.squeeze(classes[my_image_prediction])))
```

Demonstration

To demonstrate the model we save the trained parameters into a pickle file which contains the trained values of the weights and biases. The saved values can be used any time to predict the test input by passing the parameter into the predict function along with the input image. The saving of the parameters saves out time of training the model again and again which is a time consuming thing as the dataset is large and the model is huge.

To see the demonstration pick up the marker kept on the table and write any English capital letter on the paper covering the whole paper and we will take the image of the letter you write and show you the prediction of the letter by passing it into the model.

THANK YOU

~Ashish Kumar
~Satyam Agrawal
~Sarah Anwar