SAP Challenge

Kou Wen

January 11, 2018

1 Introduction

The fancy title of SAP Challenge is *Which Novel Do I Belong To?*. Translating the title into machine learning language is classifying the documents into correct categories (text classification). There are quite a mount of classification algorithms which can be applied to solve this issue, e.g. KNN, Logistic regression (For multiple classes, it is preferred as SoftMax), decision tree, linear discriminant analysis, SVM, etc. Those approaches can be regarded as linear approach which assumes the data is linear separable. But if kernel trick is preferred, that is another kind of story. The assumption is that those data is not linear separable that no appropriate hyperplane can be learnt using the above approaches, so kernel trick would seek for another feature space which can linearize the feature from the original feature space to make the problem linear separable again. But in this task, I would like to adopt nowadays fashion approach Convolutional Neural Network aka. *CNN*. In this task, I would like to tackle the different layers: convolutional layers, rectifier linear unit(ReLU) layers, pooling layers, fully connected layers, etc and manipulate the learnable parameters for example to try how many filters is suitable for each convolutional layers to seek a good CNN structure which can well solve classification problem.

In the *Which Novel Do I Belong To?*, there is another critical issue in feature extraction. The representation of each text is not natural English word so it is not possible to apply any pretrained word embedding as feature directly. But it is said ”each character has a deterministic mapping”, so each character is not random and it has its own semantic meaning. Before I build the CNN deep learning algorithm, I would like to train the character level embedding using some word embedding algorithms.

In section2, the detail of the algorithms is described. In section3, experiments and analysis are explained. In section4, conclusion is summarized.

2 Algorithm Detail

2.1 Feature Extraction

2.2 CNN architecture

The definition of the limit of $f(x)$ at $x = a$ denoted as $f'(a)$ is:

$$f'(a) = \lim_{h\to 0} \frac{f(a+h) - f(a)}{h} \tag{1}$$

The following code can be used in sage to give the above limit:

```
def illustrate(f, a):
    """
    Function to take a function and illustrate the limiting definition of a derivative at a given po
    """
    lst = []
    for h in srange(.01, 3, .01):
     lst.append([h,(f(a+h)-f(a))/h])
    return list_plot(lst, axes_labels=['$x$','$\\frac{f(%.02f+h)-f(%.02f)}{h}$' % (a,a)])
```

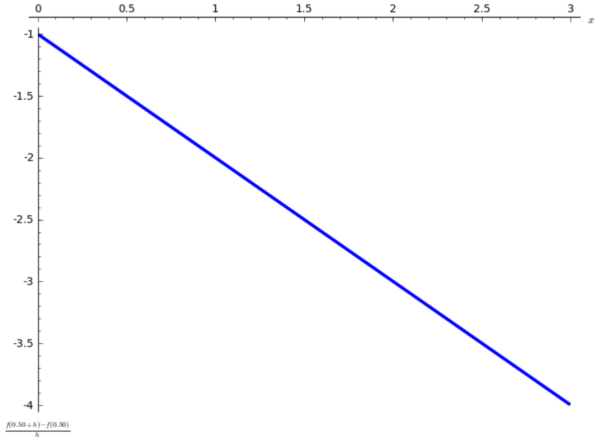

Figure 1: The derivative of $f(x) = 1 - x^2$ at $x = .5$ converging to -1 as $h \to 0$.

If we want to plot the tangent at a point α to a function we can use the following:

$$
\begin{aligned}
y =& ax + b && \text{(definition of a straight line)}\\
& f'(a)x + b && \text{(definition of the derivative)}\\
& f'(a)x + f(a) - f'(a)a && \text{(we know that the line intersects } f \text{ at } (a, f(a)))
\end{aligned}
$$

We can combine this with the approach of the previous piece of code to see how the tangential line converges as the limiting definition of the derivative converges:

```
def convergetangentialline(f, a, x1, x2, nbrofplots=50, epsilon=.1):
    """
    Function to make a tangential line converge
    """
    clrs = rainbow(nbrofplots)
    k = 0
    h = epsilon
    p = plot(f, x, x1, x2)
    while k < nbrofplots:
        tangent(x) = fdash(f, a, h) * x + f(a) - fdash(f, a, h) * a
        p += plot(tangent(x), x, x1, x2, color=clrs[k])
        h += epsilon
        k += 1
    return p
```

The plot shown in Figure 2 shows how the lines shown converge to the actual tangent to $1 - x^2$ as $x = 2$ (the red line is the 'closest' curve).

Note here that the last plot is given using the **real** definition of the derivative and not the approximation.

3 Experiment

In this report I have explored the limiting definition of the limit showing how as $h \to 0$ we can visualise the derivative of a function. The code involved `https://sage.maths.cf.ac.uk/home/pub/18/` uses the

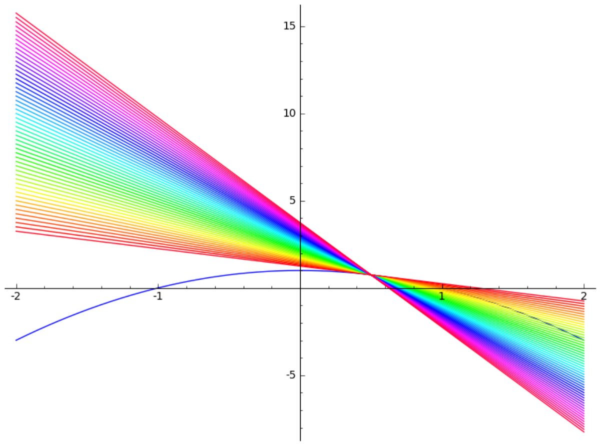

Figure 2: Lines converging to the tangent curve as $h \to 0$.

differentiation capabilities of Sage but also the plotting abilities.

There are various other aspects that could be explored such as symbolic differentiation rules. For example:

$$\frac{dx^n}{dx} = (n+1)x^n \text{ if } x \neq -1$$

Furthermore it is interesting to not that there exists some functions that **are not** differentiable at a point such as the function $f(x) = \sin(1/x)$ which is not differentiable at $x = 0$. A plot of this function is shown in Figure 3.

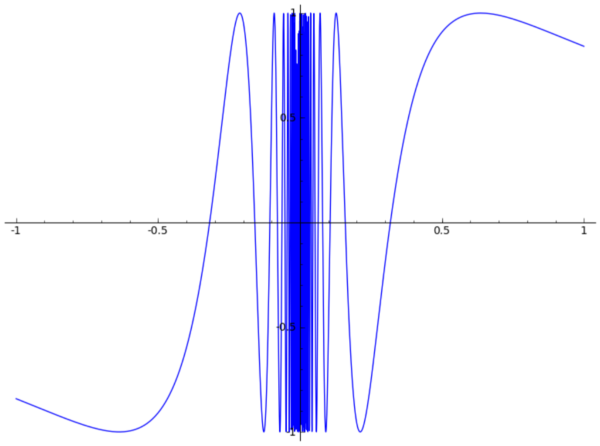

Figure 3: None differentiable function at $x = 0$.

4 Conclusion

References