NoteBook1-hs

March 30, 2018

Contents

1 Information about Grenade Library

- Grenade is a neural network written in pure Haskell for the purpose of writing fast neural networks that are concise and precise
- This library gains a lot of expressability and power from its use of `Dependent Types`.
- A Dependent type system allows the a type signature to be dependent on a value. A language like Idris is built upon this idea.
  - An example of a Dependent type would be an array in which it is a type error to even try to access an out of range element or a Tuple where the 2nd element is always greater than the first element.
- An example of a simple network written in Haskell is

```
type SampleNet = Network '[ FullyConnected 10 1, Logit ]
                         '[ 'D1 10, 'D1 1, 'D1 1 ]

randomMyNet :: MonadRandom SampleNet
randomMyNet = randomNetwork
```

  - Here we make a Network that is fully connected that takes 10 inputs and returns 1 output
  - Notice that fullyConnected takes 10 and 1, which correspond to the numbers in the 2nd list 'D10 and 'D1. This second list are the shape. Where 'D1 10 represents a 1D 10 element vector and 'D1 1 represents a 1D vector with 1 element

  - Notice we also have a logit layer which just performs a signmoid function. Also note that this type takes no term level information because it doesn't effect the shape of the network at said point.
  - So really we just made a simple network that does logisitc regression.
  - The randomMyNet is just a way to initalize with random weights.
  - Notice due to the power of the type system, there is almost no term level code that needs to be done to construct such a network

- More examples of such a network can be seen on Grenade's Github which shows a few examples most notably a MNIST netowrk with ~1.5% error and a Shakespeare `RecurrentNetwork`

2 Problems!?!?

- So I ended up doing small tweaks on the MNIST GitHub Example, However I wasn't able to properly load the MNIST data.

```
λ> x = readMNIST "./train-images-idx3-ubyte"
λ> runExceptT x
*** Exception: ./train-images-idx3-ubyte: hGetContents: invalid argument (invalid byte
```

- I ended up running out of time before figuring out how to properly format the MNIST data so I can read it. Which is a shame because the example code provides `readMNIST` and `parseMINST` along with a test I could just run. If I got that working I could have just swapped what the MNIST type was and tested many networks and see how types played with each other

- Due to this setback, I ended up just tweaking the example code slightly and staying with that