Racket for Deep Learning

Charles Earl
Data Scientist, Automattic
charles.earl@automattic.com
Contribute at https://github.com/charlescearl/DeepRacket
Outline

- Overview of deep learning
- Potential of Racket for deep learning
- Progress on the DeepRacket library
- What should the next steps be?
Overview of Deep Learning

- What are neural networks?
- What is deep learning?
- How does deep learning work?
What are neural networks?
What is deep learning?

From Nvidia
How does deep learning work?

\[
\frac{\partial}{\partial W_{ij}} J(W) = a_j^{(l)} \delta_i^{(l+1)} \\
(\text{compute gradient})
\]

\[
\delta^{(3)} = a^{(3)} - y
\]

\[
\delta^{(2)} = (W^{(2)})^T \delta^{(3)} \cdot \frac{\partial g(z^{(2)})}{\partial z^{(2)}} \\
(\text{error term of the hidden layer})
\]

from https://sebastianraschka.com/faq/docs/visual-backpropagation.html
Potential of Racket for Deep Learning

• How does functional programming apply to deep learning?

• Specifying deep networks dynamically.

• Specifying and running in Racket
Recurrent Nets

\[ h_t = W \cdot h_{t-1} \]
\[ o_t = \tanh(h_t + U \cdot x_t) \]
Thinking of Deep Learning Functionally
(Colah, 2015 & Balduzzi 2016)

Encoding RNN

RNN

from Neural Networks, Types, and Functional Programming
http://colah.github.io/posts/2015-09-NN-Types-FP/
As dynamic graph computation

Figure 4: An example of a computation graph for $g(x, j) = \text{tanh}(W_1 x + b) + \text{tanh}(W_2 e_j + b)$, and the corresponding code.

from Dynet, [https://github.com/clab/dynet](https://github.com/clab/dynet)
Specifying and running in Racket

(let ([W (def-filter (3 2))]
       [x (def-tensor (2))])
  ...
  [sgd (create-stock-grad-optimizer )]
  [rnn (def-expr (+ (* W x) (* U h) b))])

(for ([i (range epochs)])
  (compute-fwd rnn)
  (compute-backward rnn (loss yhat y))
  (compute-backward rnn)
  (update-weights sgd rnn)))
Benefits

• Giving students exposure to deep learning
• Exploration of new functional architectures
• New paradigms for machine learning (e.g. alternatives to backprop)
• CUDA/OpenCL access for Racket
Progress on the DeepRacket library
DeepRacket

• A cudnn wrapper
  • Built with typed Racket, Math and FFI libraries
  • Specification of RNNs and forward (estimator) and backward (gradient) computation
  • Still need loss and optimization (SGD)!

• A dynet wrapper
  • Specify selected networks
  • Training
A quick DeepRacket demo
What’s next?
Useful directions?

• Better Syntax!

• Device agnostic
  • GPU and CPU
  • OpenCL

• Dynamic graph computation
https://github.com/charlesearl/DeepRacket