Racket for Deep Learning

Charles Earl Data Scientist, Automattic <u>charles.earl@automattic.com</u>

Contribute at <u>https://github.com/</u> <u>charlescearl/DeepRacket</u>

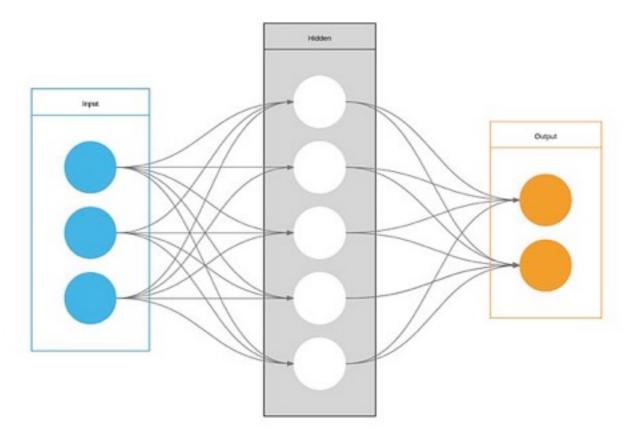
Outline

- 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?

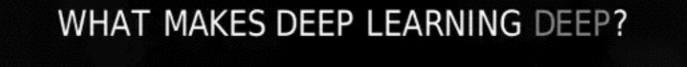
What are neural networks?



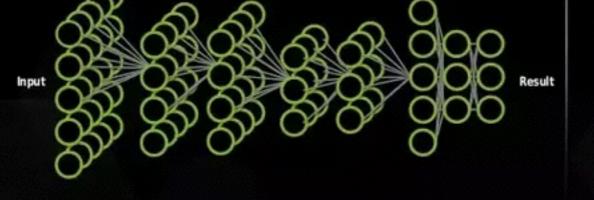


By LearnDataSci - www.learndatasci.com, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=49369360

What is deep learning?







Today's Largest Networks

-10 layers 1B parameters 10M images -30 Exaflops -30 GPU days

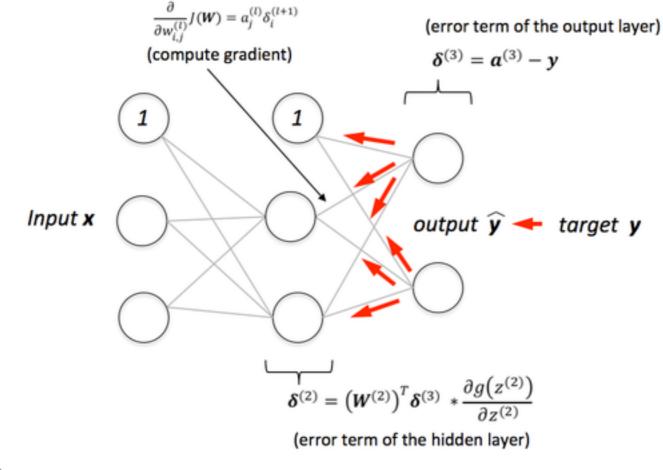
Human brain has trillions of parameters - only 1,000 more.

IN IDIA

AUTOMATTIC We are passionate about making the web a better place.

From Nvidia

How does deep learning work?





from https://sebastianraschka.com/faq/docs/visual-backpropagation.html

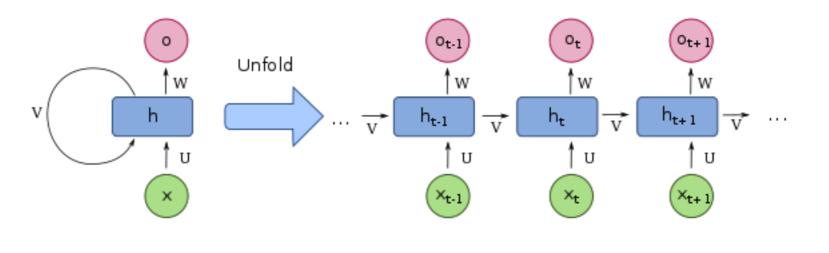
Potential of Racket for Deep Learning

How does functional programming

apply to deep leaning?

- Specifying deep networks dynamically.
- Specifying and running in Racket

Recurrent Nets



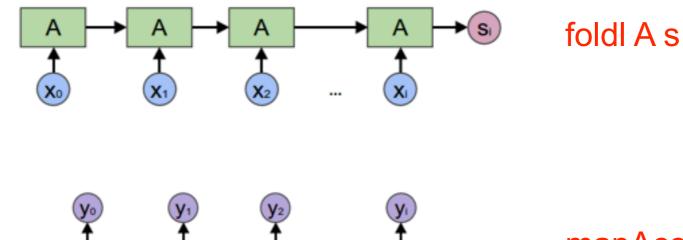
$$h_t = W \cdot h_{t-1}$$
$$o_t = tanh(h_t + U \cdot x_t)$$



By François Deloche - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=60109157

Thinking of Deep Learning Functionally (Colah, 2015 & Balduzzi 2016)

Encoding RNN



RNN

mapAccumR a s

AUTOMATTIC We are passionate about making the web a better place.

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

А

(Si

X

As dynamic graph computation

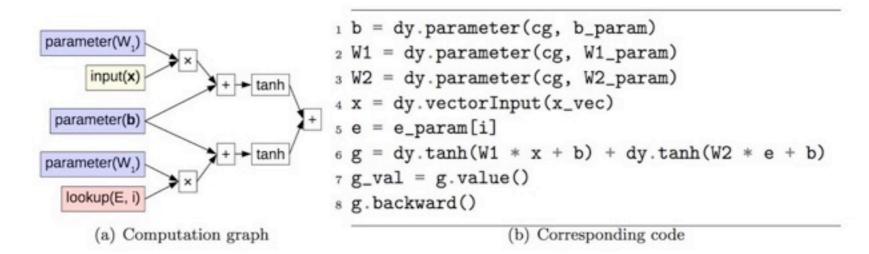


Figure 4: An example of a computation graph for $g(x, j) = \tanh(W_1 * x + b) + \tanh(W_2 * e_j + b)$, and the corresponding code.

AUTOMATTIC We are passionate about making the web a better place.

from Dynet, https://github.com/clab/dynet

Specifying and running in Racket

(let ([W (def-filter (3 2))] [x (def-tensor (2))] Specify the network ... [sgd (create-stock-grad-optimizer)] [rnn (def-expr (+ (*W x) (* U h) b))]) (for ([i (range epochs)]) Now train it! (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

	run-dynet.rkt - DrRacket				
run-dynet.rkt# (define)#	Check Syntax Dev	Debug 😂 🖂	Macro Stepper 🐲 🛃	Run 🔛	Stop
#lang racket					
require					
"simple-dynet-api.rkt")					
(begin					
(init_dynet)					
(let+					
(
[hidden-size 8]					
<pre>[cg (get_computation_graph)]</pre>					
<pre>[pc (get_parameter_collection)]</pre>					
[sgd (get_simple_sgd pc)]					
<pre>[w (add_parameters_shape_two cg 8 2 pc)]</pre>					
<pre>[q (add_parameters_shape_two cg 8 2 pc)]</pre>					
[yval (get_dynet_vector 1)]					
[xval (get_dynet_vector 2)]					
<pre>[xval_ptr (get_dynet_vect_ptr xval)]</pre>					
[yval_ptr (get_dynet_vect_ptr yval)]					
[y (create_outputs cg yval)]					
<pre>[x (create_n_inputs_vtr cg xval 2)]</pre>					
<pre>[v (add_parameters_shape_two cg 1 8 pc)]</pre>					
<pre>[b (add_parameters_shape_one cg 8 pc)]</pre>					
<pre>[a (add_parameters_shape_one cg 1 pc)]</pre>					
[h (create_tanh w x b)]					
[pred (create_pred v h a)]					
[loss_expr (create_loss pred y)]					
(loss 0.0)					
1					
(for ([epoch (in-range 30)])					
(for* ([mi (in-range 4)])					
(let+					
<pre>[x1 (modulo mi 2)]</pre>					
<pre>[x2 (modulo (quotient mi 2) 2)]</pre>					
[input1 (if (eq? x1 1) 1.0 -1.0)]					
[input2 (if (eq? x2 1) 1.0 -1.0)]					
[output (if (not (eq? x2 x1)) 1.0 -	1.011				
(set_dynet_vptr_xval_ptr 0 input1)					
(set_dynet_vptr_xval_ptr 1 input2)					
(set_dynet_vptr yval_ptr 8 output)	loss averill				
(set! loss (+ loss (get_scalar_loss cg	tuss_expr////				
(do_backward_loss cg loss_expr)					
(update_parans sgd 1.0))) (set! loss (/ loss 4.0))					
(display (format "Current loss is: ~a\n" 1	04411111				

AUTOMATTIC

What's next?

Useful directions?

- Better Syntax!
- Device agnostic
 - GPU and CPU
 - OpenCL
- Dynamic graph computation

https://github.com/ charlescearl/DeepRacket