NPFL116 Compendium of Neural Machine Translation

Notes on Deep Learning

March 1, 2017

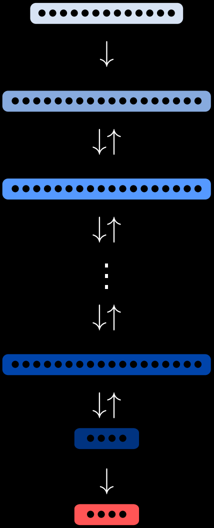
Jindřich Libovický, Jindřich Helcl

Charles Univeristy in Prague
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics

Deep Learning

- machine learning that hierarchically infers suitable data representation with the increasing level of complexity and abstraction (Goodfellow et al.)
- formulating end-to-end relation of a problems’ raw inputs and raw outputs as parameterizable real-valued functions and finding good parameters for the functions (me)
- industrial/marketing buzzword for machine learning with neural networks (backpropaganda, ha, ha)

Neural Network



$$x$$

$$\downarrow$$

$$h_1 = f(W_1 x + b_1)$$

$$\downarrow$$

$$h_2 = f(W_2 h_1 + b_2)$$

$$\downarrow$$

$$\vdots$$

$$\downarrow$$

$$h_n = f(W_n h_{n-1} + b_n)$$

$$\downarrow$$

$$o = g(W_o h_n + b_o)$$

$$\downarrow$$

$$E = e(o, t) \quad \rightarrow \quad \frac{\partial E}{\partial o}$$

$$\vdots$$

$$\uparrow$$

$$\vdots$$

$$\uparrow$$

$$\vdots$$

$$\uparrow$$

$$\vdots$$

$$\uparrow$$

$$\frac{\partial E}{\partial W_o} = \frac{\partial E}{\partial o} \cdot \frac{\partial o}{\partial W_o}$$

$$\uparrow$$

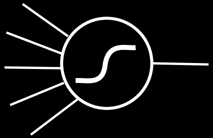
$$\frac{\partial E}{\partial o}$$

Building Blocks (1)

- individual neurons / more complex units like recurrent cells *(allows innovations like inventing LSTM cells, ReLU activation)*
- libraries like Keras, Lasagne, TFSlim conceptualize on layer-level *(allows innovations like batch normalization, dropout)*
- sometimes higher-level conceptualization, similar to functional programming concepts *(allows innovations like attention)*

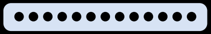
Building Blocks (2)

Single Neuron



- computational model from 1940’s
- adds weighted inputs and transforms to input

Layer



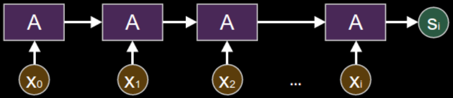
$$f(Wx + b)$$

...*f* nonlinearity, *W* ...weight matrix, *b* ...bias

- having the network in layers allows using matrix multiplication
- allows GPU acceleration
- vector space interpretations

Encoder & Decoder

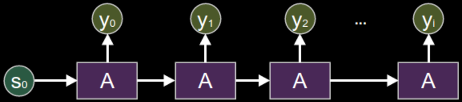
Encoder:



Functional fold (reduce) with function

`foldl a s xs`

Decoder:



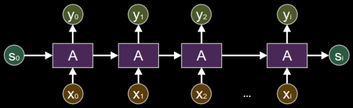
Inverse operation – functional unfold

`unfoldr a s`

Source: Colah's blog (http://colah.github.io/posts/2015-09-NN-Types-FP/)

RNNs & Convolutions

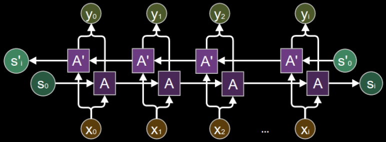
General RNN:



Map with accumulator

`mapAccumR a s xs`

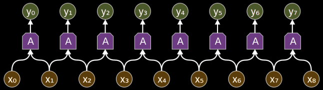
Bidirectional RNN:



Zip left and right accumulating map

```
zip (mapAccumR a s xs)
(mapAccumL a' s' xs)
```

Convolution:



Zip neighbors and apply function

`zipWith a xs (tail xs)`

Source: Colah’s blog (http://colah.github.io/posts/2015-09-NN-Types-FP/)

Optimization

- data is constant, treat the network as function of parameters
- the differentiable error is function of parameters as well
- clever variants of gradient descent algorithm

Deep Learning as Alchemy

- there no rigorous manual how to develop a good deep learning model – just rules of thumb
- we don’t know how to interpret the weights the network has learned
- there is no theory that is able to predict results of experiments (as in physics), there are only experiments

Recoding in mathematics

Algebraic equations

$$\begin{aligned} 10x^2 - x - 60 &= 0 \\ 0.2x^3 - 2x^2 - 10x + 4 &= 0 \\ -2x^2 - 10 &= 0 \end{aligned}$$

...became planar curves

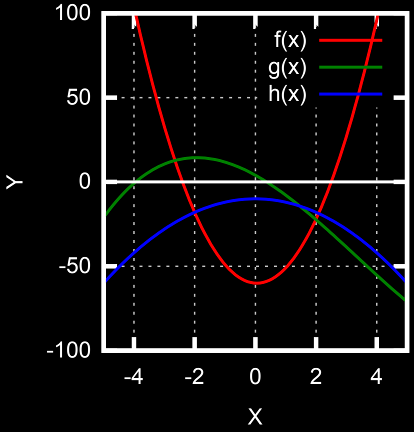
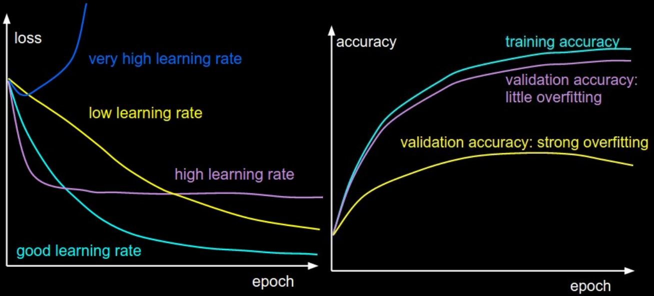


Image: Existential comics (`http://existentialcomics.com/`)

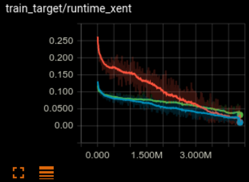
Watching Learning Curves

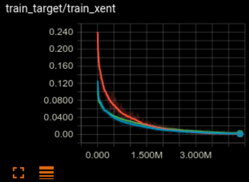


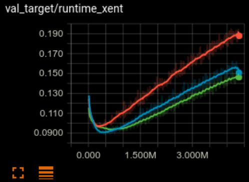
Source: Convolutional Neural Networks for Visual Recognition at Stanford University
(`http://cs231n.github.io/neural-networks-3/`)

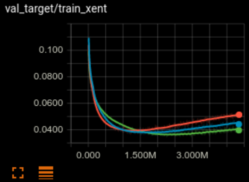
Other Things to Watch During Training

- train and validation loss

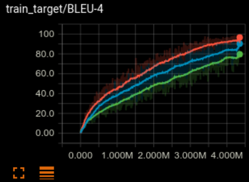


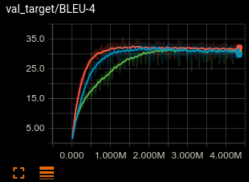






- target metric on training and validation data





MT is hard

- language are not word-by-word equivalent
- there is not better way of expressing the sentence than the language itself
- even if we have a system, it’s hard to evaluate it

What’s Strange on Neural MT

- we naturally think of translation in terms of manipulating with symbols
- neural network represents everything as real-space vectors

Reading for the Next Week

Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." *Advances in neural information processing systems*. 2014.
`https://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf`

Question:
What are the problems of the presented architecture? How do you think the neural MT continued after publishing this paper?