Introductory Notes on Machine Translation and Deep Learning

Jindřich Helcl, Jindřich Libovický

February 26, 2020

LANGTECH

EUROPEAN UNION
European Structural and Investment Fund
Operational Programme Research,
Development and Education

Charles University
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics

What is Machine Translation?

What is machine translation?

Time for discussion...

What We Think...

- MT does not care what translation is
- We believe people know what translation is and that it is captured in the data
- We evaluate how well we can mimic what humans do when they translate

What is Deep Learning?

Deep Learning in Context of AI

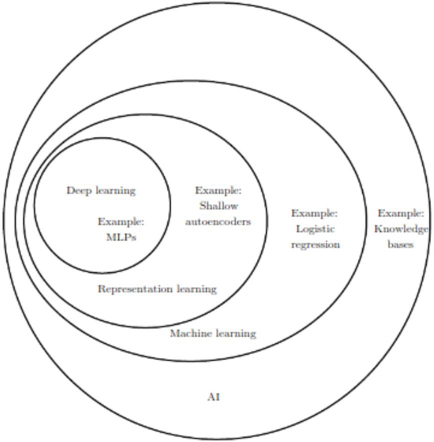


Figure 1.4: A Venn diagram showing how deep learning is a kind of representation learning, which is in turn a kind of machine learning, which is used for many but not all approaches to AI. Each section of the Venn diagram includes an example of an AI technology.

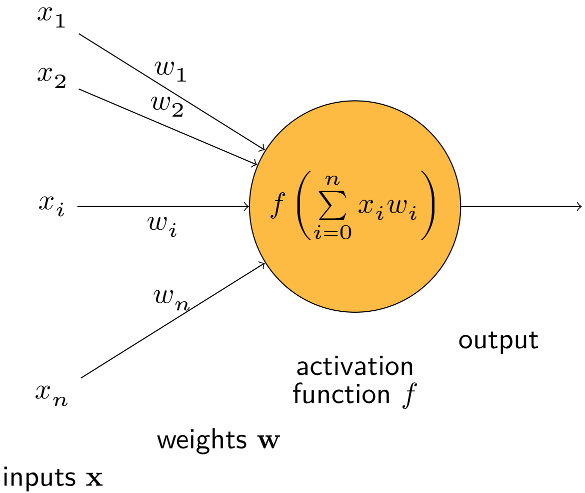
Source: Goodfellow et al., *Deep Learning Book*, `www.deeplearningbook.org`

Deep Learning

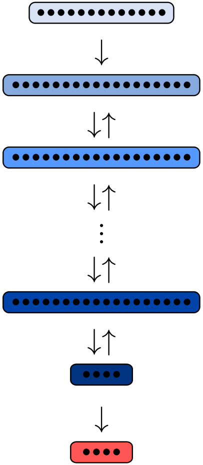
- Machine learning that hierarchically infers suitable data representation with the increasing level of complexity and abstraction (Goodfellow et al., 2017)
- Formulating end-to-end relation of a problems' raw inputs and raw outputs as parameterizable real-valued functions and finding good parameters for the functions (JL, 2017)
- Industrial/marketing buzzword for machine learning with neural networks (backpropaganda, ha, ha)

Deep Learning as Mathematics

Single Neuron



Neural Network



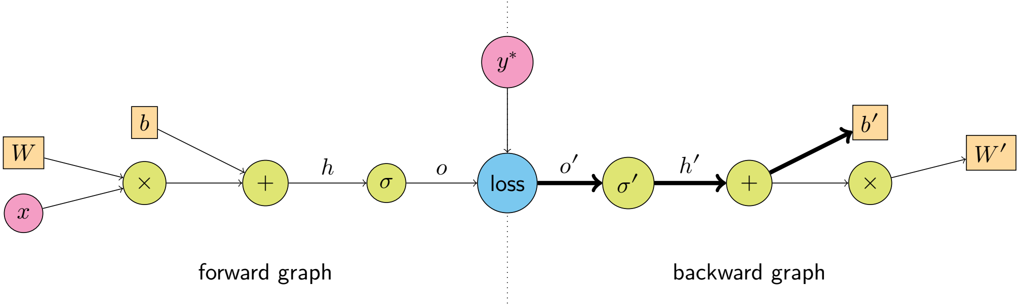
$$\begin{tikzcd}
x \arrow[d] & & \\
h_1 = f(W_1 x + b_1) \arrow[d] & & \vdots \\
h_2 = f(W_2 h_1 + b_2) \arrow[d] & & \vdots \arrow[u] \\
\vdots \arrow[d] & & \vdots \arrow[u] \\
h_n = f(W_n h_{n-1} + b_n) \arrow[d] & & \vdots \arrow[u] \\
o = g(W_o h_n + b_o) \arrow[d] & & \frac{\partial E}{\partial W_o} = \frac{\partial E}{\partial o} \cdot \frac{\partial o}{\partial W_o} \arrow[u] \\
E = e(o, t) \arrow[rr] & & \frac{\partial E}{\partial o} \arrow[u]
\end{tikzcd}$$

Implementation

Logistic regression:

$$y = \sigma(Wx + b) \tag{1}$$

Computation graph:

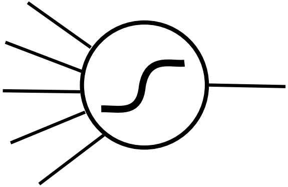


Building Blocks (1)

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

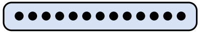
Building Blocks (2)

Single Neuron



- Computational model from 1940’s
- Adds weighted inputs and transforms to input

Layer



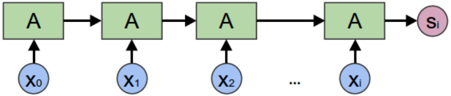
$$f(Wx+b)$$

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

- Having the network in layers allows using matrix multiplication
- Allows GPU acceleration
- Vector space interpretations

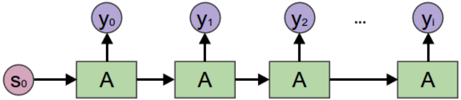
Encoder & Decoder

Encoder:



Functional fold (reduce) with function
`foldl a s xs`

Decoder:

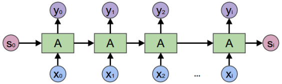


Inverse operation – functional unfold
`unfoldr a s`

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

RNNs & Convolutions

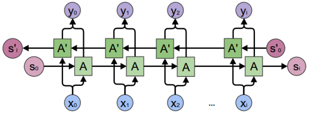
General RNN:



Map with accumulator

`mapAccumR a s xs`

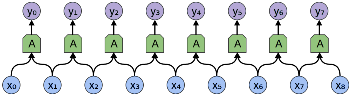
Bidirectional RNN:



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

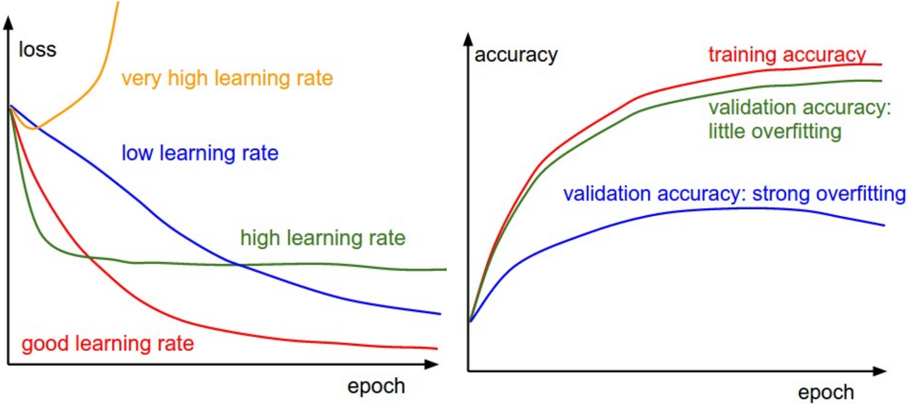
- Data is constant, network is treated as function of parameters
- Differentiable error is function of parameters as well
- Clever variants of gradient descent algorithm

Deep Learning as Alchemy

Deep Learning as Alchemy

- No rigorous manual for developing a good deep learning model – just rules of thumb
- Unclear how to interpret the weights the network has learned
- No theory that is able to predict results of experiments (as in physics), there are only experiments

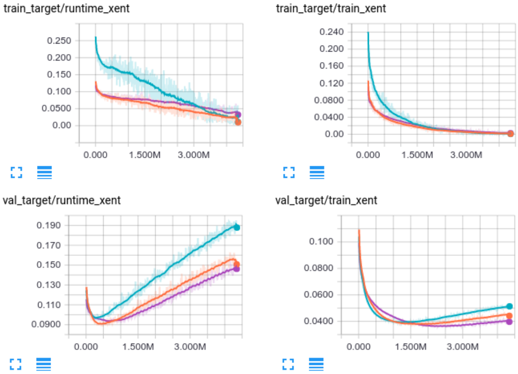
Watching Learning Curves



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

Other Things to Watch During Training (1)

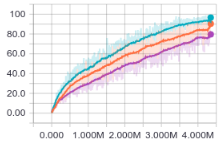
- Train and validation loss



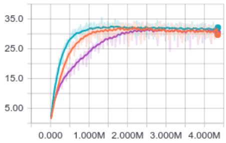
Other Things to Watch During Training (2)

- Target metric on training and validation data

train_target/BLEU-4

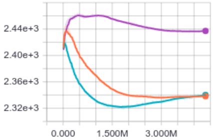


val_target/BLEU-4

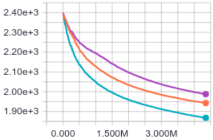


- L2 and L1 norm of parameters

train_l1

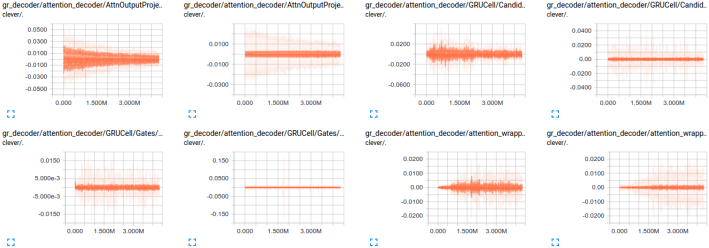


train_l2



Other Things to Watch During Training (3)

- Gradients of the parameters



- Non-linearities saturation

Machine Translation and Deep Learning

What's Strange on Neural MT

- We naturally think of translation in terms of manipulating with symbols
- Neural networks represent everything as real-space vectors
- Ignore pretty much everythng we know about language

Reading for the Next Week

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. “Deep learning.” *Nature* 521.7553 (2015): 436.
`http://pages.cs.wisc.edu/~dyer/cs540/handouts/deep-learning-nature2015.pdf`

Question:
Can you identify some implicit assumptions the authors make about sentence meaning while talking about NMT? Do you think they are correct? How do the properties that the authors attribute to LSTM networks correspond to your own ideas how should language be computationally processed?