Neural networks

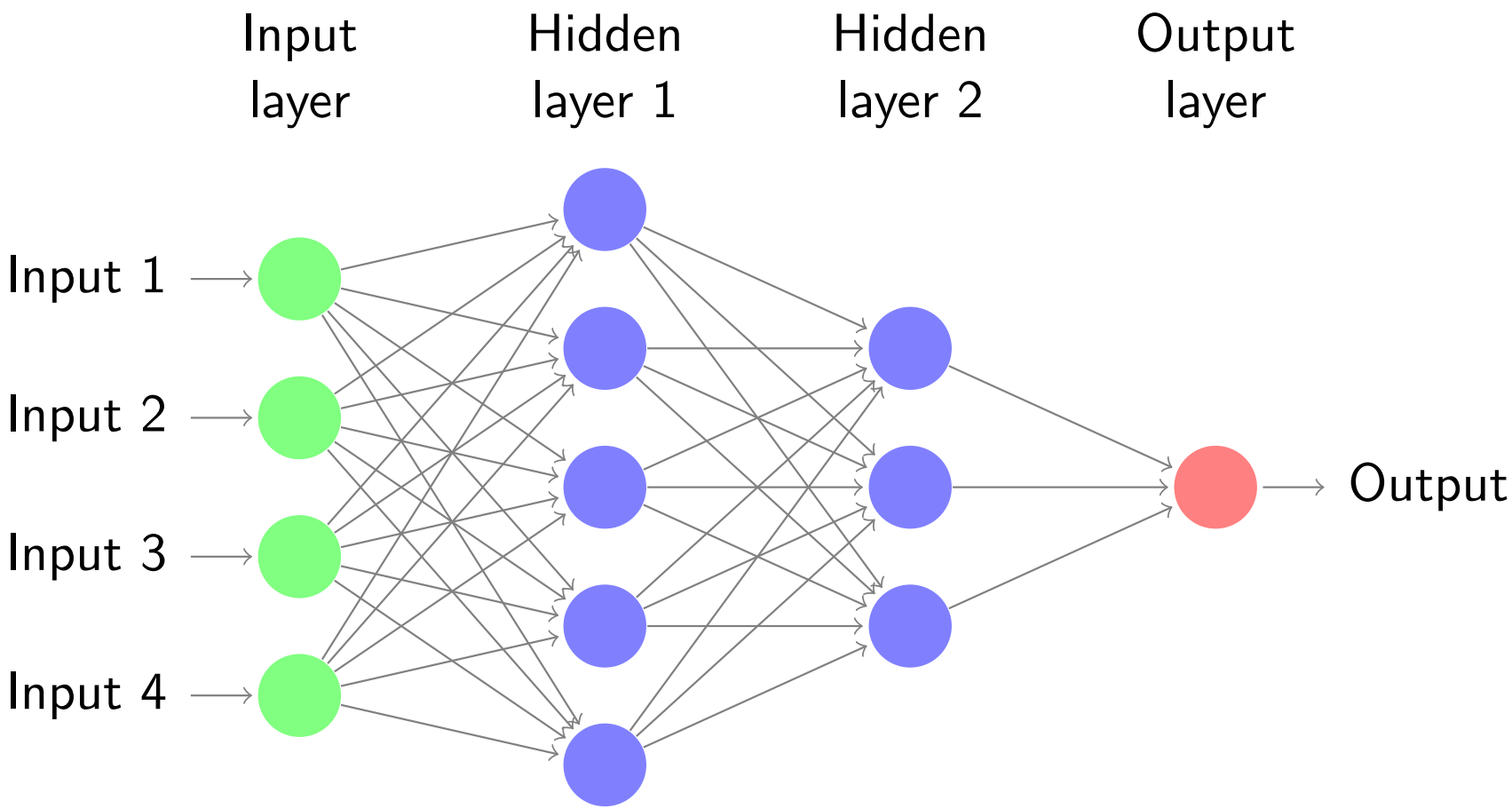
Architectures and training tips

Sebastian Björkqvist

IPRally Technologies

09.01.2019

What is a neural network?



Modified from `http://www.texample.net/tikz/examples/neural-network/`

What is a neural network?

At each hidden layer node i the output value is calculated by

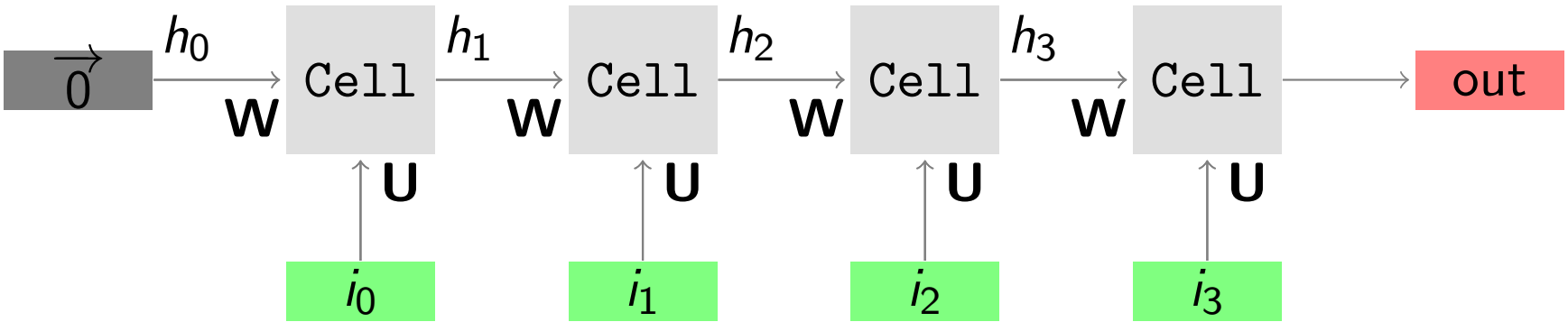
$$o_i = \sigma(\sum w_{ki} o_{ki-1} + b_i).$$

The function σ is called the activation function. It must be non-linear to allow the network to learn non-linear dependencies.

Why neural networks?

- Can approximate any function [Hornik, 1991]
- May learn to respond to unexpected patterns
- Useful especially when the amount of data is large
- Less need for feature engineering compared to traditional ML methods

Recurrent neural network (RNN)



Processes each element of the input sequence in order, and keeps information about the past elements in a hidden state vector.

Recurrent neural network (RNN)

At each timestep t the new hidden state is calculated using the new input at this timestep and the existing hidden state. The most basic version is the following:

$$h_t = \sigma(Wh_{t-1} + Ui_t + b).$$

Other RNN architectures (for instance LSTM or GRU) use more complicated ways of updating the hidden state to control the flow of information to and from the hidden state.

RNN pros and cons

+ Accepts input of variable size, i.e. sequences (time series, sentences etc)
+ May learn long-term dependencies
– Training may be slow when sequence length is large
– Can be hard to train

Convolutional neural network (CNN)

TODO: Picture here
Extracts features of two-dimensional input (usually an image) using convolutional and pooling layers.

CNN pros and cons

+ Works well with image data
– Pre-existing models can be fine-tuned for specific tasks
– Does not take into account position or orientation of the object

Challenges when training neural networks

- Finding the optimal neural network layout is often time-consuming
- The model may be sensitive to changes in hyperparameters
- A model may take several hours or even days to train

Tips and tricks

- Write simple tests for your model [Roberts, 2017]
  - Check that each layer actually changes weights
  - Make sure that model converges on tiny data set
- Stick to well-known architectures when starting out (e.g. LSTM/GRU for sequential data)
- Start by using small batch size
  - Usually makes model less sensitive to other hyperparameters
- Use normalization (batch, layer, group...)
  - Start by trying batch normalization for CNN and Feed-forward nets and layer normalization for RNN

References


Nielsen, Michael A. *Neural Networks And Deep Learning*. Determination Press, 2015. http://neuralnetworksanddeeplearning.com/


Hornik, Kurt. *Approximation Capabilities of Multilayer Feedforward Networks*. Neural Networks, 4(2), 251–257, 1991.


Roberts, Chase. *How to unit test machine learning code. Medium.com 2017. https://medium.com/@keeper6928/how-to-unit-test-machine-learning-code-57cf6fd81765.*