



UNIVERSITY  
*of York*

DEPARTMENT OF ELECTRONICS  
BIOLOGICALLY INSPIRED COMPUTATION

## Neural Networks For Speech Interaction

### **Abstract**

The history and technology behind speech based human computer interfaces,  
and the role neural networks play in current implementations.

Zak West

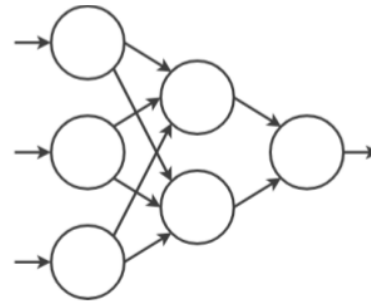
May 13, 2019

## What is a neural network



### What is a neural network?

- Made up of Neurons (Nodes)
- Takes inputs
- Multiple accumulate with weights
- Transfer function
- Often trained with backpropagation
- Often using labeled data
- Can have loops, to recurrent networks




Neural networks are a type of machine learning, pioneered in the 1940s by McCulloch and Pitts.

They are made up of multiple neurons (or nodes), that are structured into layers. Each neuron has one or more inputs and a set of weights associated with them. These neurons operate by applying a simple process to their inputs to create an output. Each neuron takes its inputs and multiplies them by the associated weights and then applies a transfer function. This transfer function may be a binary step, sigmoid, or tanh to name a few.

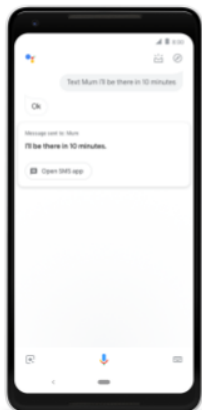
The thing that makes Neural networks special is that they can be trained, or taught. This can be done in many ways, but the simplest is using labelled data and backpropagation. This approach uses the correct output values and alters the weights of the neurons to get the network to agree.

Neural networks can have this simple structure, but they often have more complex structures. such as backwards connections from one layer to the previous. This is known as recurrence and is one tool used for speech recognition. They can also be modified to accept complex inputs like audio or even images, this normally done using convolution neural networks.

## History of Voice based AI assistants




### History of Voice based AI assistants



assistant.google.com

- Pioneered by IBM in 1960's
- Advanced by DARPA
- Siri versions launched in 2011
- From a Hidden Markov Model to Neural networks 2014
- Microsoft in 2014, Amazon 2014, Google in 2016

IBM Shoe Box



[http://sysrun.haifa.il.ibm.com/ibm/history/exhibits/specialprod1/specialprod1\\_7.html](http://sysrun.haifa.il.ibm.com/ibm/history/exhibits/specialprod1/specialprod1_7.html)

Speech-based human-computer interfaces have a surprisingly long history. Which starts off around the '50s, below some of these are listed.

In 1953 Bell Labs created a system called Audrey, which could recognise spoken digits.[9] It used the differences in resonance that different vowel sounds produce to classify digits.[1]

In 1961 IBM unveiled a device at the Seattle worlds fair called the “shoebox”[12], This device was, as you may have guessed, about the size of a shoe box. It allowed people to perform arithmetic using voice commands by saying digits and commands words such as “plus” or “minus”. It would then illuminate the correct light as an output. It was implemented using 3 filters and discreet logic.

In the '70s DARPA created a tool called Harpy that could recognise around a thousand words. [3]

By the '80s this team had developed the system to not only recognise words but phrases, using hidden Markov models [5]. These models can be thought of as a kind of probabilistic Finite state machine.

In the '90s this technology started to appear for personal computers, with the main players being Microsoft and IBM.

But it wasn't till 2011 and the introduction of Siri[13] to the iPhone that this technology became widespread. Siri was originally developed by a spin-out company from DARPA (again) and it still used hidden Markov models. It wasn't until 2014 that neural networks were used for Siri. [15]. Siri now uses many advanced ANN techniques such as CNNs, DNNs and RNNs.

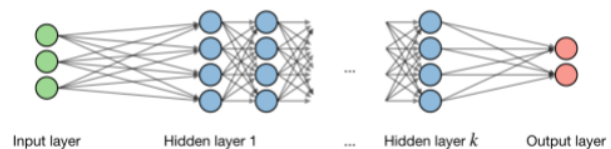
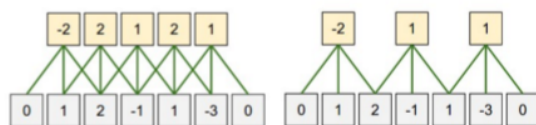
Now there are many competitors to Siri. Such as Microsoft's Cortana or Amazon's Alexa.

## CNN and DNN



### CNN and DNN

- CNN
  - Takes set/stream inputs
  - Applies a function
  - Creates new set/stream
  - Essentially a filter
- DNN
  - Many layer networks
  - Each layers with many neurons
  - 1000's neurons



Now, most voice assistants are using Convolution neural networks on time series inputs.[7]

Convolution itself is the process of taking multiple input variables (in this case the last few audio samples) and applying a function ( aka a kernel) to them to get a new waveform. A simple convolution in images is to detect edges. You know you have an edge if the pixel to your left is black and the one to your right is white. For audio a simple convolution can be low pass filter, the convolution function make the current output the average of the last few inputs. This removes fast changes, thus removing high frequencies. These can also provide a time delay to a signal.

In general, CNNs tend to have multiple of these filters. With each one being trained to apply a different filter to the input.

These may also change the sample rate of the data as it gets further into the network because more complicated features span longer amounts of time. A phoneme has a longer duration than the frequencies that make it up and a word takes longer than the phonemes that make it up. Changing this sample rate allows you to have a lot fewer neurons in later layers, which make the networks smaller and faster.

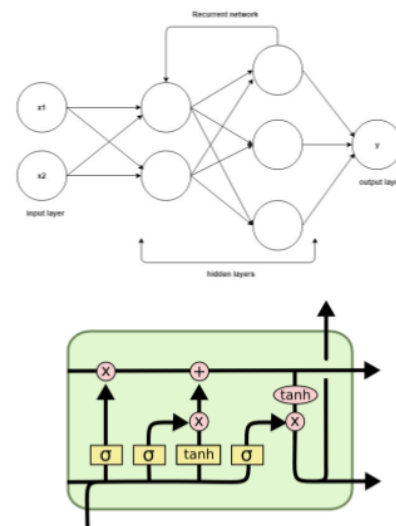
DNNs are deep neural networks. It wasn't long after the first neural networks were proposed that multilayer networks were created, in 1965 the first paper was published on them [2]. Here deep refers to many layers. Modern deep neural networks can have 100s of layers and potentially thousands of nodes. This allows for the networks to represent more complex and higher order functions. As well as higher levels representing more complex ideas. Deep neural networks can also contain many different types of network. I.e they could have an input convolutional network then a feed forward. Or an LSTM and then a feedforward

## RNN and LSTM



### RNN and LSTM

- Most networks are feed forward
- Cannot use past inputs
- Speech is time series, needs past information
- LSTM better than RNN
- Can 'choose' what to remember



RNNs are recurrent neural networks. This means there is some form of backwards connection in the network. This means that the previous state of the network can affect the next state. These were first proposed by John Hopfield in 1982 [4]. This is a very useful feature to have when working with time series data like speech. As it is very common for the previous inputs to affect the current state. Unfortunately, RNNs have a problem. Feedback. This causes vanishing or exploding gradients and outputs. Which make them very hard to train. RNNs also constantly update their state with the new input, this means they're not very good when something from a long time ago should influence the current state. I.e a couple of words back or the previous sentence.

A new architecture of RNNs was proposed in 1997[6] called LSTM, which was then improved in 1999[8] LSTMs fix this by allowing the network to choose how much it remembers of the current state by using multiple gates for each unit. This means each LSTM cell can remember only important cues. This also fixes the gradient problem.

LSTM have been shown to be very good at a wide array of time series prediction and classification tasks, including dealing with voice.

## Natural language processing



### Natural language processing

- Once you have converted speech to text you need to do something with it
- Command words
- More ANN's
  - Word2vec



Once you have converted that speech to text you have to do something with it. This is where natural language processing comes in. The simplest way of doing this is using command words. This is what the IBM shoe box from earlier did. In that case, the command words were "plus" or "add". This is still used, for simple phrases like "turn the music down". But that isn't very natural for people and it would be hard to enumerate every possible command.

The next step is using rules based approaches. These don't depend on exact matches, but still tend to have limited numbers of known phrases. Phrases such as "schedule a meeting at 4pm with John" can be done with rules based approaches. [14]

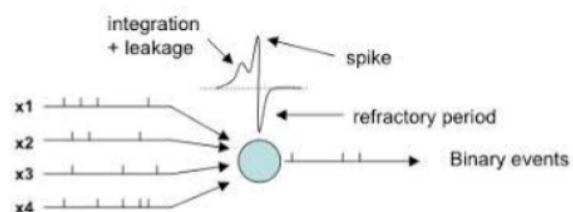
Since the '80s ANNs are also used to do this. This is a statistical approach to NLP. Currently LSTMs are often used because they are good with sequences. Representing words in an ANN is quite a difficult task there are many methods but the most popular at the moment is Word2Vec which uses vectors to map to words. [10]

## Future?



### Future?

- Custom hardware making AI faster
- More data being collected
- New architectures applied?
  - Spiking Neural Networks
  - Stacking Neural Networks



Currently, custom devices are being created and sold, whose primary function is speech interaction. These are the Amazon Echo, Google Home and the Apple HomePod.

These devices currently use traditional computer hardware. Most likely ARM CPUs. But since they are single-purpose devices they could start to use custom chips with dedicated ANN and DSP functionality. The speed improvements from this would be massive and so networks could be made more complex.

There is also interest in using spiking neural networks and other models based more directly off of real neurons[11]. The idea is that these networks are continuous in the time domain. This is maybe an advantage when processing continuous time domain data like speech.

## References

- [1] KH Davis, R Biddulph, and Stephen Balashek. “Automatic recognition of spoken digits”. In: *The Journal of the Acoustical Society of America* 24.6 (1952), pp. 637–642.
- [2] Aleksei Grigor’evich Ivakhnenko and Valentin Grigorévich Lapa. *Cybernetic predicting devices*. CCM Information Corporation, 1965.
- [3] B. T. Lowerre. “The Harpy speech recognition system”. PhD thesis. Carnegie-Mellon Univ., Pittsburgh, PA., Apr. 1976.
- [4] John J. Hopfield. “Neural networks and physical systems with emergent collective”. In: (1982).
- [5] Lawrence R Rabiner and Biing-Hwang Juang. “An introduction to hidden Markov models”. In: *ieee assp magazine* 3.1 (1986), pp. 4–16.
- [6] Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780. URL: [http://www.cs.bham.ac.uk/~pjh/sem1a5/pt1/pt1\\_history.html](http://www.cs.bham.ac.uk/~pjh/sem1a5/pt1/pt1_history.html).
- [7] LeCun, Yann and Bengio, Yoshua. “The Handbook of Brain Theory and Neural Networks”. In: ed. by Arbib, Michael A. Cambridge, MA, USA: MIT Press, 1998. Chap. Convolutional Networks for Images, Speech, and Time Series, pp. 255–258.
- [8] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. “Learning to forget: Continual prediction with LSTM”. In: (1999).
- [9] Biing-Hwang Juang and Lawrence R Rabiner. “Automatic speech recognition—a brief history of the technology development”. In: *Georgia Institute of Technology. Atlanta Rutgers University and the University of California. Santa Barbara* 1 (2005), p. 67.
- [10] Tomas Mikolov et al. “Efficient estimation of word representations in vector space”. In: *arXiv preprint arXiv:1301.3781* (2013).
- [11] Wei-Yu Tsai et al. “Always-on speech recognition using truenorth, a reconfigurable, neurosynaptic processor”. In: *IEEE Transactions on Computers* 66.6 (2017), pp. 996–1007.
- [12] URL: [https://www.ibm.com/ibm/history/exhibits/specialprod1/specialprod1\\_7.html](https://www.ibm.com/ibm/history/exhibits/specialprod1/specialprod1_7.html).
- [13] URL: <https://www.apple.com/siri/>.
- [14] P J Hancox. URL: [http://www.cs.bham.ac.uk/~pjh/sem1a5/pt1/pt1\\_history.html](http://www.cs.bham.ac.uk/~pjh/sem1a5/pt1/pt1_history.html).
- [15] Steven Levy. URL: <https://www.wired.com/2016/08/an-exclusive-look-at-how-ai-and-machine-learning-work-at-apple/>.