**NEURAL NETWORKS**

**SIRI**

The convergence of artificial intelligence, smartphone adoption and availability of a huge amount of consumer data is leading to a new generation of virtual assistants. Wearables have a crucial role as well: speech recognition is now built into every major operating system, allowing users to speak with the machine.  
Despite an army of scientists devoted their lives to this challenge for decades, if you think about Siri app, it seems we are still far away from **the dream of speaking conversationally to a machine**. The good news is that technology is improving fast and the future virtual assistants will be able to put your words into the proper context and reply accordingly.  
The task is much more complex than you think. In this post I’m going to explain why and envision future developments. Machines talking with people come from far away. In 2003 DARPA hugely invested in a five year, 500 persons project aimed at building a virtual assistant. The government wanted to develop software to help military commanders with communication optimization. This helper was named CALO, the Cognitive Assistant that Learns and Organizes. Siri is then the progeny of the largest artificial intelligence project in U.S. history and has been brought to life by 3 scientists who launched a standalone iPhone app called Siri in early 2010. Several weeks after launch, they received a phone call which, I assume, sounded like this: “Hey, it’s Steve. What are you doing tomorrow? Want to come over to my house?” It was Steve Jobs and Apple acquired the technology for a reported $150 to $250 million in 2011. The problem is that Siri is also the orphan of Steve… he died the day after Siri debuted.

In Siri, Apple’s personal assistant, mainly used technology is Speech recognition software.

**Phase 1: voice recognition**  
It’s apparently the easy part, but it’s where everything begins, so it can’t be trivial. When you give Siri a command, your device collects your analog voice, convert it in an audio file (it’s translated into binary code) and send it to Apple servers. The nuances of your voice, the noise around and the local expressions make difficult to get it done right. It’s called **Human User Interface** versus the standard Graphical User Interface we are used to. It’s important here that, everyday, Apple collects millions of queries of people speaking multiple languages, in many accents, while living on different continents. In other words with their actions and mistakes, people are contributing to the largest crowd sourced speech recognition experiment ever tried on earth. Siri app today receives roughly a billion requests per week and Apple states its speech recognition capability has just a 5 percent word error rate. Last year Apple acquired the speech recognition company Novauris Technologies, a spinoff of Dragon Systems and also hired several speech recognition experts, to get to this point.

SIRI's main tasks, at a high level, involve:

1. Using ASR (Automatic speech recognition) to transcribe human speech (in this case, short utterances of commands, questions, or dictations) into text.
2. Using natural language processing (part of speech tagging, noun-phrase chunking, dependency & constituent parsing) to translate transcribed text into "parsed text".
3. Using question & intent analysis to analyze parsed text, detecting user commands and actions.  ("Schedule a meeting", "Set my alarm", ...)
4. Using data mashup technologies to interface with 3rd-party web services such as OpenTable, WolframAlpha, to perform actions, search operations, and question answering (Utterances SIRI has identified as a question, that it cannot directly answer, it will forward to more general question-answering services such as WolframAlpha)
5. Transforming output of 3rd party web services back into natural language text (eg, Today's weather report -> "The weather will be sunny")
6. Using TTS (text-to-speech) technologies to transform the natural language text from step 5 above into synthesized speech.

In Speech recognition software, Error back propagation neural network is used.

In the proposed system neural network training is based on the calculation of epoch of the audio signal and then used these epoch value for the training of the neural network. The epoch values are basically instant of significant excitation of the vocal-tract system during production of speech. The input data sets used to train the neural network can be partitioned in to the independent epochs. Each epoch representing a

temporal value of the input data. Back propagation neural network used in the system in following steps.

1. First we choose and fix the architecture for the network, which comprised of input, hidden and output units, all units will contain their sigmoid functions value.

2. Then we will assign the weights among all the processing units. The assignments of weights are random and usually between -0.5 and 0.5.

3. Each input pattern is used in order to re-train the weights in the neural network.

4. Next calculating each epoch for input audio data, a termination condition is checked

In neural network architecture the weights of input and hiddenlayers are adjusted according to the target output values [11] The input data is considered as E which is proliferate through the topology so that we can easily note all the observed results Oi(E) for the output units Oi. Meanwhile, we also note all the observed values hi(E) for the hidden units. After that, for each output unit Ok, we calculate its epoch as follows:

The epoch values from the output node are helpful to compute epoch value for the hidden nodes. This method is termed as Back Propagation because here we inseminate this data backwards by using the network [12]. For each hidden node

Hk, we compute the epoch as follows: Here, we take the epoch value for each output node and then multiply it by the weight of the network from hidden Hk to the output nodes. After that we add all these simultaneously and multiply the summation by hk(E)\*(1 - hk(E)). When we get all the calculated epoch values related with every unit (hidden and output), finally pass this information into the weight changes Δij among units i and j. The calculation is defined as: for all weights wij between input node Ii and hidden node Hj, and summation of all units are as:

￼

The Back Propagation learning process requires the following components:

1. Set of input data patterns, input values and target output values.

2. Learning Rate Value.

3. Algorithm termination criteria.

4. Procedure for the updating weights.

5. Nonlinear functions or sigmoid functions.

6. Initial random weight values.

￼

**In proposed system Back propagation neural architecture contains the all above units and the basic steps of proposed system are defined as follows.**

1. Read the input audio Signal.

2. Extract the epoch values

3. Train the neural network on the basis of epoch values.

4. Applied the back propagation neural network for the classification.

5. Matching the input data with the trained data.

6. Recognized the input.

￼