### Study of Neural Networks Applied to Speech Recognition

A close up of a piece of paper

Description automatically generatedNeural Networks

**CNN vs RNN**

They are both different architecture’s of neural nets that perform well on different types of data. RNNs are good with series of data (one thing happens after another) and are used a lot in problems that can be framed as “what will happen next given…” while CNNs are especially good at problems like image classification.

RNNs (recurrent neural networks) are made up of one node. It is fed data then outputs a result back into itself, and continues to do this. Breakthroughs like LSTM (long short term memory) make it smart at remembering things that have happened in the past and finding patterns across time to make its next guesses make sense.

CNNs (convolutional neural networks) essentially have three parts, convolution layers, pooling layers, and fully-connected layers. It usually takes a 2D (sometimes more dimensions) matrix and outputs a result.

Convolution starts at the top left and takes a small window with a certain width and height and performs an operation on that, the operation is usually a matrix multiplication where the matrix to multiply by is decided via gradient descent to get the best final results. It then moves according to a stride parameter and does the same. It does this all the way across the image and outputs a new image.

Pooling is similar in the sense that it breaks the image down using small windows; however, the operation it runs on this small window is usually (average, max, or min) to combine the small window into a single pixel.

After a set amount of convolutions and pooling, the final output is put through a fully connected layer, which is a conventional feed forward neural network to output a result.

You can think of the pooling and convolution layers as a form of image pre-processing similar to what was done in traditional computer vision, except the parameters like the matrix in each convolution layer is decided by gradient descent.

**Convolutional Neural Networks**

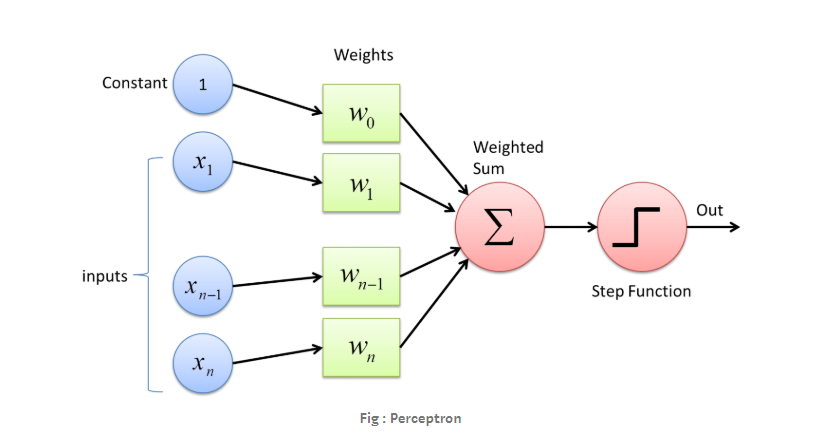
https://github.com/lukas/ml-class/blob/master/videos/cnn-audio/preprocess.py

**RNN**

<https://github.com/cmaroti/speech_recognition>

Neural networks used in this experiment

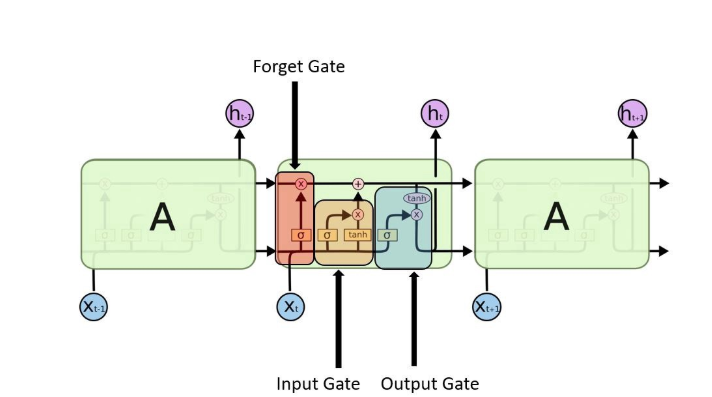
Perceptron



2D Convolution



LSTM



Performance Conclusions

