**Case Study on application of neural networks—**

**“Classification of Emails”**

This case study is built to classify a given mail as spam (not good) or ham (good) email. This will help you not read any unnecessary emails.

The email will be classified based on its content.

**What is spam email?**

Spam email is unsolicited and unwanted junk email sent out in bulk to an indiscriminate recipient list. Typically, spam is sent for commercial purposes. It can be sent in massive volume by botnets, networks of infected computers. Spam email is unsolicited and unwanted junk email sent out in bulk to an indiscriminate recipient list. Typically, spam is sent for commercial purposes. It can be sent in massive volume by botnets, networks of infected computers.

**What is the purpose of spam email?**

The main idea of spam is to make a profit. Mass email sending is cheap and easy compared to physical mail. Near 250 billion emails are sent globally every day, and [45%](https://www.spamlaws.com/spam-stats.html) of them are spam. Even though the [conversion rate](https://sendpulse.com/support/glossary/email-conversion-rate) is shallow — one reply in 12,500,00 spam emails sent, [according to TechRadar](https://www.techradar.com/news/internet/computing/spam-gets-1-response-per-12-500-000-emails-483381) — it is still worth a [spammer's](https://sendpulse.com/support/glossary/spammer) while.

In most cases, spammers disguise unwanted emails with the mask of promotions offering something from the “ultimate diet drug” to “100% working remedy for men’s health”. Aside from commercials, the most popular types of spam include adult content and emails promising financial independence by following simple instructions.

According to the [Register of Known Spam Operations (ROKSO)](https://www.spamhaus.org/rokso/), most of the world’s spam activity comes from about a hundred of so-called “spam gangs” operating mainly from the USA, Russia, and China.

**Why do we get spam emails?**

1. Your email address was exposed during a data leakage
2. Your email was harvested by a specialized tool
3. Your email was stolen from someone’s contact list
4. Your email was randomly generated
5. A dishonest company sold your email

**Recurrent Neural Network:**  
The fundamental feature of a Recurrent Neural Network (RNN) is that the network contains at least one feed-back connection, so the activations can flow round in a loop. That enables the networks to do temporal processing and learn sequences, e.g., perform sequence recognition/reproduction or temporal association/prediction. Recurrent neural network architectures can have many different forms. One common type consists of a standard Multi-Layer Perceptron (MLP) plus added loops. These can exploit the powerful non-linear mapping capabilities of the MLP, and also have some form of memory. Others have more uniform structures, potentially with every neuron connected to all the others, and may also have stochastic activation functions. For simple architectures and deterministic activation functions, learning can be achieved using similar gradient descent procedures to those leading to the back-propagation algorithm for feed-forward networks.  
   
In sequential tasks such as natural language and speech processing, there is always dependence of present input data upon the previous applied inputs. Task of RNNs is to find the relationship between current input and the previous applied inputs. In theory RNNs can make use of information sequence of any arbitrarily length, but in practice they are limited to looking back only a few steps.

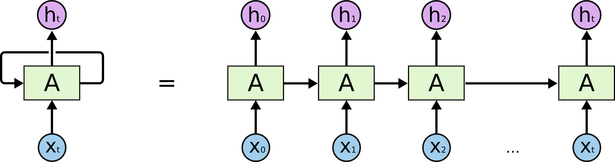


Figure: Basic architecture of Recurrent Neural Networks

   
The above figure shows a RNN being unfolded into a full network. By unfolding we simply mean that we are repeating the same layer structure of network for the complete sequence.  
Xt is the input at time step t. Xt is a vector of any size N.  
A is the hidden state at time step t. It’s the “memory” of the network. It is calculated based on the previous hidden state and the input at the current step.  
Represented by At= f (W Xt +U At-1)  
Here W and U are weights for input and previous state value input. And f is the non-linearity applied to the sum to generate final cell state.  
  
One of the appeals of RNNs is the idea that they might be able to connect previous information to the present task, such as using previous video frames might inform the understanding of the present frame. If RNNs could do this, they’d be extremely useful. Sometimes, we only need to look at recent information to perform the present task. For example, consider a language model trying to predict the next word based on the previous ones. If we are trying to predict the last word in “the clouds are in the *sky*,” we don’t need any further context – it’s pretty obvious the next word is going to be sky. In such cases, where the gap between the relevant information and the place that it’s needed is small, RNNs can learn to use the past information. In theory, RNNs are absolutely capable of handling such “long-term dependencies.” A human could carefully pick parameters for them to solve toy problems of this form. Sadly, in practice, RNNs don’t seem to be able to learn them.

**Conclusion:**

So we will first implement it using simple RNN. After that we will try to increase its accuracy.