

- Neural Turing machines utilise external memory
- The RNN reads and writes to multiple memory positions and uses an attention distribution to determine the extent to which it manipulates each position
- The attention distribution is determined using the combination of their similarity to a query vector, the attention distribution of the previous step, and a shift filter allowing its attention to move relatively
- This allows them to learn storing long sequences in memory and looping over it and mimic a lookup table or number sorter
- Attentional Interfaces learns to evaluate sets of information to determine which parts are most relevant and focusing on them.
- These can take the output from a neural network and return an input of the most relevant data for the network to look out for.
- Adaptive Computation Time allows for each recurrence of the RNN to perform a different number of computations.
- This allows you to find a balance between the performance of the neural network and the computation time.
- Neural Programmers learns to create programs to solve tasks it can't solve itself, like arithmetic.
- The programs are generated one operation at a time and act upon the outputs of previous operations
- The neural net runs multiple possible operations at once and averages them weighted by the probability that we ran that operation
- Then we can define a loss function that adjusts these probabilities and trains the neural net to create programs.

<https://distill.pub/2016/augmented-rnns/>