Improving Differentiable Neural Computers Through Memory Masking, De-allocation, and Link Distribution Sharpness Control

Róbert Csordás
robert@idsia.ch

Jürgen Schmidhuber
juergen@idsia.ch

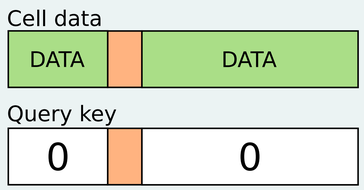
IDSIA — Scuola universitaria professionale della Svizzera italiana SUPSI — Università della Svizzera italiana

Summary

- DNC is very general and can solve many tasks. But task-specific methods often outperform it
- Can we close the gap between DNC and task-specific methods?
- We improve DNC's address generation methods
  1. **Masked content-based lookup** avoids score calculation problems of the cosine distance comparator
  2. **Modified deallocation** avoids aliasing through non-erased memory contents
  3. **Sharpness enhancement** of temporal linkage addresses overcomes their noisiness

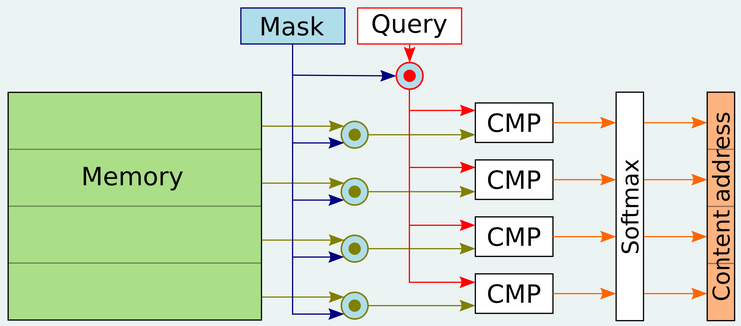
Content-based lookup

- Goal: retrieve unknown information (the cell data) based on partial knowledge (the query key)
- Content based lookup compares a query key to each memory cell to produce an address distribution
- The score is normalized by the whole memory contents, thus the effect of **unknown data** (green) can **dominate the score** calculation



$$D(\mathbf{u},\mathbf{v}) = \frac{\mathbf{u}\cdot\mathbf{v}}{|\mathbf{u}||\mathbf{v}|}$$

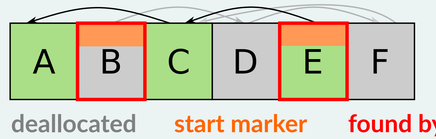
- Solution: explicit **masking** of both the data and the query key



- Advantages:
  - **Dynamic** key-value separation
  - More **general** than key-value memory
  - The decision what to search for can be made after storing
  - Can be used for any attention mechanism

Deallocation problem

- Allocation states are tracked by usage counters
- Memory allocation chooses the least used address
- Freeing memory is achieved by decreasing the usage counters of previously read addresses
- Problem: memory contents do not change. **Content-based lookup still can find the deallocated cells**



Which is correct?
BDFC?
ECA?

- Solution: **erase** the memory while decrementing usage counters

$$\boldsymbol{M}_t = \boldsymbol{M}_{t-1}\left[\odot\, \psi_t \mathbf{1}^T\right] \odot \left(\boldsymbol{E} - \mathbf{w}_t^w \mathbf{e}_t^\intercal\right) + \mathbf{w}_t^w \mathbf{v}_t^\intercal$$

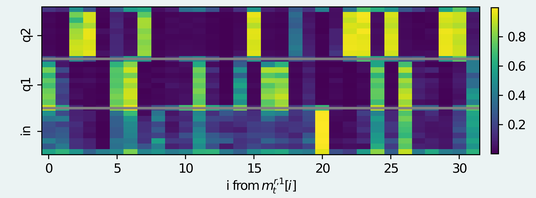
Link sharpness control

- **Noise** from write address distributions is **accumulating** in the link matrix
- Address distributions resulting from temporal linkage might **not sum to 1**
- Solution: **exponentiation and renormalization**

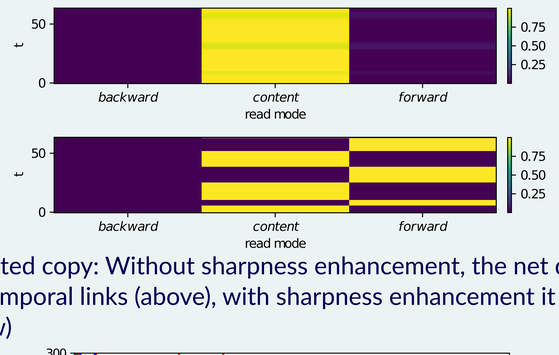
$$\mathbf{f}_t^i = S\left(\boldsymbol{L}_t \mathbf{w}_{t-1}^{r,i}, s_t^{f,i}\right) \qquad \mathbf{b}_t^i = S\left(\boldsymbol{L}_t^\intercal \mathbf{w}_{t-1}^{r,i}, s_t^{b,i}\right)$$

$$S(\mathbf{d},s)_i = \frac{(\mathbf{d}_i)^s}{\sum_j (\mathbf{d}_j)^s}$$

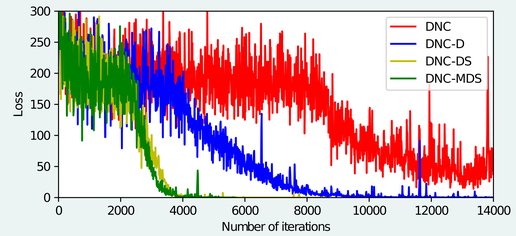
Results



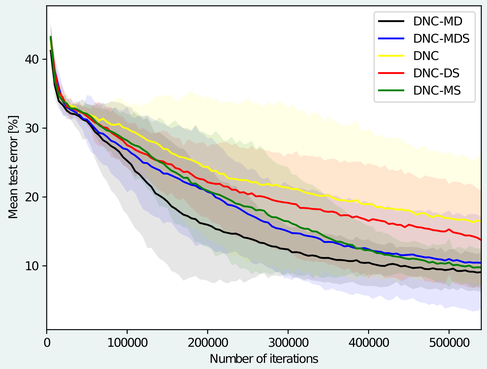
- The net actively changes the mask when looking up different parts of the stored vectors



- Repeated copy: Without sharpness enhancement, the net does not use temporal links (above), with sharpness enhancement it does (below)



- Impact of deallocation and sharpness enhancement on repeated copy task



- Mean test error on **bAbI**. Our model shows a **43%** relative **imporvement** in bAbI mean error (9.5% compared to 16.7%)

GitHub/xdever/dnc