* a "library" of RNNs

dx
dy
mag
ori
ori hist
shift image (dx, dy)
translational symmetry
point symmetry (corners)
FEP
rotation symmetry
perceptron classifier —— learns

* composition mechanism
* execution capability
* set of text images + ground truth (12x11 ~~3-15x15~~) see $NOTES/Tango/chars
* classification experiments

---

library inputs

w

Library Model (RNN) set

~~name~~

RNN(1).name = 'dx'

RNN(1).W = W

RNN(1).f = f-vector $\equiv \begin{bmatrix} 'f1' \\ 'f2' \\ \vdots \end{bmatrix}$

RNN(1).u0 = initial vector

inputs + outputs 0

RNN(1).I = $[I_1, I_2, \ldots I_m]$

RNN(1).O = $[O_1, O_2, \ldots O_k]$

RNN(1).uf = $\bar{u}_f$

num_~~iters~~ steps

no change

.type = 'program'

'input'

~~Execution Model~~

interface function

out $= TA_run_RNN($input name, RNN_name & Library ~~ids~~ ids$)$.

input:

weight matrix, W : $n \times n$ array

function at each node, f_i

input ~~neurons (identify)~~ m values

inputs neurons $\mathcal{I} = \{ I_1, \ldots I_m \}$ $I_i \in \{1, 2, \ldots n\}$

initial configuration, $\bar{u}_0$

end of computation configuration or max steps $\bar{u}_f$

output neurons (identify) $\mathcal{O} = \{O_1, O_2 \ldots O_k\}$

$O_i \in \{1, 2, \ldots n\}$

execution

$$\bar{u}_{t+1} = W \bar{u}_t$$

$$\bar{u}_{t+1} = \bar{f}(\bar{u}_{t+1}) \equiv [f_i(u_{t+1,i})]$$

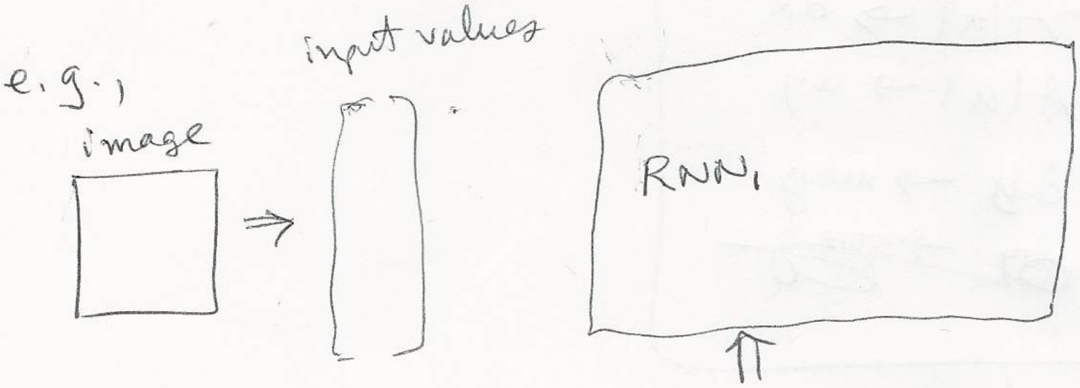
$$\text{till } \bar{u}_{t^*} == \bar{u}_f$$

output

values of $[u_{f,O_1}, \ldots u_{f,O_k}]$

~~given~~

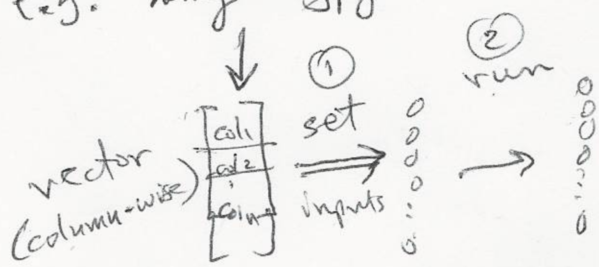
Composition Mechanism



function selection
+ composition

$\rightarrow RNN_i$ ⮡

e.g. imga3.jpg → dx



input RNN

RNN(

connectivity

$RNN_i \rightarrow RNN_{o_1}$

$\quad \hookrightarrow RNN_{o_2}$

$\quad \vdots$

$\quad \hookrightarrow RNN_{o_p}$

static vs. dynamic input

↳ assume static for our problem

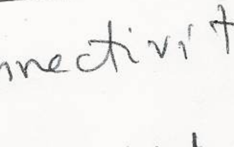
input → dx → mag

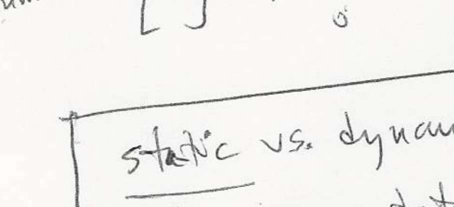
→ dy ↗

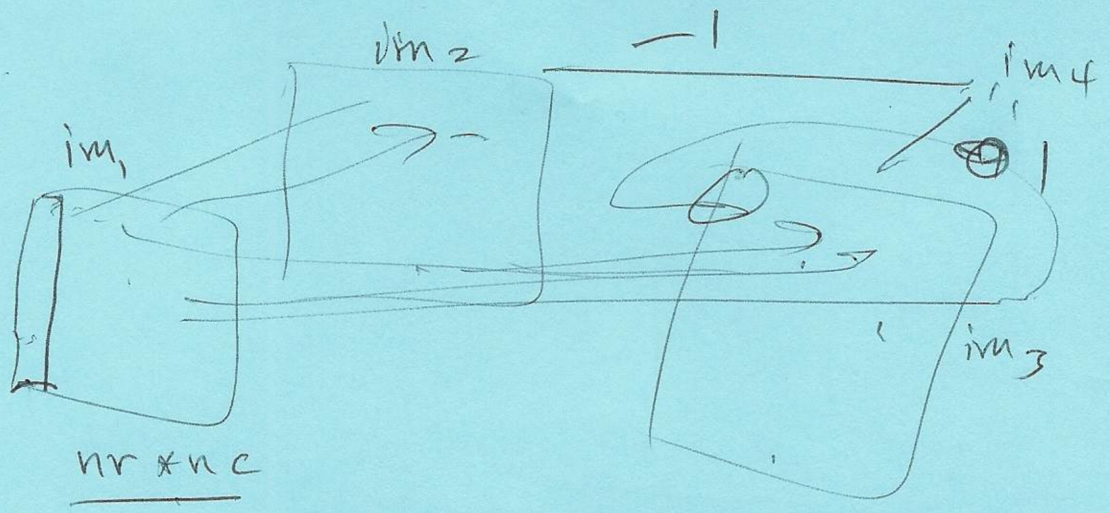
input special name

input → RNN

Alal → {dx, dy} → mag





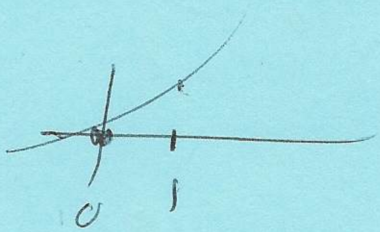


funct RNN = TA_dx_RNN(nr, nc)

%

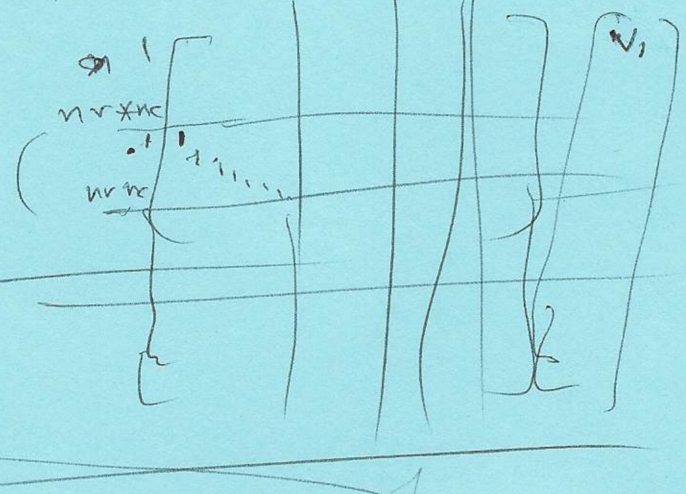
lim2 ← lim1

lim3 (index(1,1)) ← lim1 (index(1, nc))

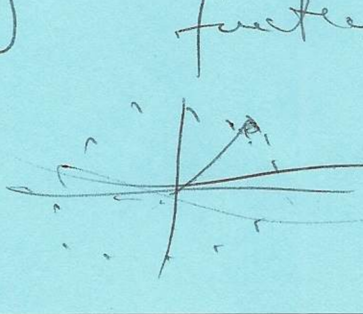


−1	1

lim4(index(1,1)) ← lim2(index(1,1)) − lim3(index(1,1))



W functions identity



0 1 2

−1	0	1	dx
−1	0	1	dy