RECURRENT NETWORKS

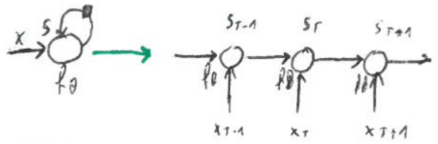
FOR SEQUENCES / VARIABLE LENGTH DATA. CONTEXT-AWARE. PARAMETER SHARING ACROSS TIME STEPS (SAME NEURON) • WHY? BECAUSE IF DIFFERENT NET FOR DIFFERENT SEQUENCE LENGTHS WITH SEPARATE PARAMS IT'S MORE EXPENSIVE AND WON'T GENERALIZE

GRAPH UNFOLDING

$S_T = f_\theta(S_{T-1}, X_T) \longrightarrow g(X_T, X_{T-1}, \dots, X_{T-N})$

• S MIGHT DEPEND ON ARBITRARY SUBSET OF X_Ts, USUALLY LOSSY

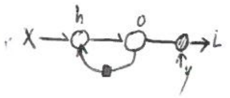
• TOUGHEST SHIT → AUTOENCODERS BECAUSE WE WANT TO RECOVER ORIGINAL SEQUENCE



• USEFUL ABSTRACTION FOR FLOWING INFORMATION FORWARD IN TIME (OUTPUTS, LOSSES) AND BACKWARD (GRADIENTS)

• **HIDDEN RECURRENCE:** UNIVERSAL APPROXIMATION MACHINE FOR DISCRETE SEQUENCES. ANY FN COMPUTABLE BY TURING MACHINE CAN BE COMPUTED BY RNN OF FINITE SIZE → PENDING IN/OUT DISCRETIZATION TO BINARY SEQUENCES AND UNBOUNDED PRECISION FLOATS.

• **OUTPUT RECURRENCE:** EASIER TO TRAIN BECAUSE NO BACKPROPAGATION THROUGH TIME IS REQUIRED. THE ONLY STATE-CARRYING INFORMATION IS THE PREVIOUS PREDICTION. LESS POWERFUL. ASSUMPTION STRONG. APPROPRIATE WHEN FULL SYSTEM STATE IS OBSERVED AND PROVIDES A TARGET. STATE MUST BE RICH ENOUGH TO 'CARRY' SUMMARY OF PAST. **TEACHER FORCING:** BACK-FED INPUTS ARE ACTUAL TARGETS, NOT OUTPUTS. → MAY YIELD POOR GENERALIZATION. MIX OUTPUTS AND ACTUAL TARGETS DURING TRAINING. **GENERATIVE RNNS:** OUTPUTS FED-BACK



HIDDEN RECURRENT EQUATIONS

$$\begin{cases} a_t = b + W s_t + U x_t \\ s_t = \tanh(a_t) \\ o_t = c + V s_t \\ p_t = \text{softmax}(o_t) \end{cases} \qquad L(x,y) = \sum_t L_t = \sum_t -\log p_{t,y_t}$$

GRADIENT COMPUTATION

BACKPROPAGATION THROUGH TIME. FOREACH a WE HAVE TO ~~PROPAGATE~~ COMPUTE $\nabla_a L$ RECURSIVELY, THROUGH GRADIENT AT FOLLOWING NODES.

1 • AT FINAL LOSS: $\frac{\partial L}{\partial L_T} = 1$

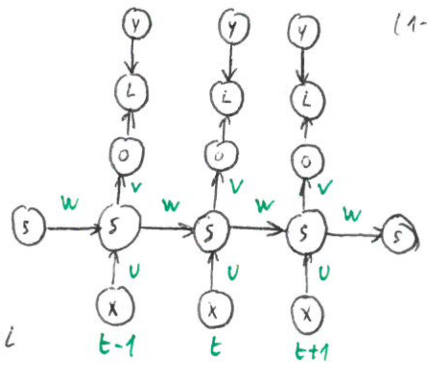
2 • $(\nabla_{o_t} L)_i = \frac{\partial L}{\partial o_{ti}} = p_{t,i} - 1_{i,y_t}$ GRADIENT ON OUTPUTS AT t, FOR ALL i

3 • $\nabla_{s_T} L = \nabla_{o_T} L \frac{\partial o_T}{\partial s_T} = \nabla_{o_T} L V$ START FROM END OF SEQUENCE T. O_T ONLY DESCENDANT

4 • $\nabla_{s_t} L = \nabla_{s_{t+1}} L \frac{\partial s_{t+1}}{\partial s_t} + \nabla_{o_t} L \frac{\partial o_t}{\partial s_t}$
$= \nabla_{s_{t+1}} L \,\text{DIAG}(1 - s_{t+1}^2) W + \nabla_{o_t} L V$

ITERATE BACK THROUGH TIME DOWN TO $t=1$. s_t HAS s_{t+1} AND o_t AS DESCENDANTS. $(1 - s_{t+1}^2)$ IS TANH'

5 •
$$\begin{cases} \nabla_c L = \sum_t \nabla_{o_t} L \frac{\partial o_t}{\partial c} = \sum_t \nabla_{o_t} L \quad \text{OUTPUT BIAS} \\ \nabla_b L = \sum_t \nabla_{s_t} L \frac{\partial s_t}{\partial b} = \sum_t \nabla_{s_t} \text{DIAG}(1 - s_t^2) \\ \nabla_V L = \sum_t \nabla_{o_t} L \frac{\partial o_t}{\partial V} = \sum_t \nabla_{o_t} L s_t^T \\ \nabla_W L = \sum_t \nabla_{s_t} L \frac{\partial s_t}{\partial W} = \sum_t \nabla_{s_t} L \,\text{DIAG}(1 - s_t^2) s_{t-1}^T \end{cases}$$



• PARAMETER GRADIENTS OBS $\nabla_{s_t} L$ IS THROUGH ALL PATHS FROM s_t TO L

RNN AS DGMs

WHAT ARE LOSSES? FOR PREDICTION, WHY NOT ESTIMATION OF CONDITIONAL DISTRIBUTION OF $y_{t+1} | y_{1..t}$? WE MAY ALSO CONDITION ON OTHER INPUTS. → FULL JOINT ACROSS WHOLE SEQUENCE. IF NO CONDITION ON y → y OUTPUTS ARE CI GIVEN SEQUENCE. IF NO LATENT STATE VARIABLES PARAMETRIZATION IS VERY INEFFICIENT. WITH S NODES JOINT PARAMETRIZATION WRT TIME IS CONSTANT; ELSE BLOWS UP EXPONENTIALLY. S_t SUMMARIZES. DECOUPLES PAST FROM FUTURE; GRAPH ONLY LOCALLY CONNECTED. OPTIMIZATION OF SHARED PARAMS MIGHT BE DIFFICULT. PARAM SHARING ⟷ CONDITIONAL IS STATIONARY, MARKOV PROPERTY, → SAME MODEL FOR DIFFERENT LENGTHS. LIKELIHOOD BECOMES $P(X) = \prod P(X_T | f_\theta(S_{T-1}, X_T))$. f_θ MAY BE LEARNED ITSELF. IN GENERATIVE MODE X_T IS SAMPLED FROM OUTPUT CONDITIONAL AND FED BACK FOR PRODUCING FURTHER STEPS. **STOP SIGNAL:** EOS SYMBOL, EXPLICIT MODELING OF T NUMBER, AS INPUT

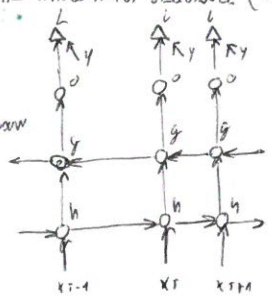
CONDITIONED SEQUENCE MODELING

WHEN WE CONDITION DISTRIBUTION ON OTHER VARS $P(y | \omega = f(x))$ • **FIXED SIZE:** EXTRA INPUT, AS INITIAL STATE. **X, Y INDEPENDENT** → CAUSAL RELATIONSHIP BETWEEN X AND PREDICTED Y → CAN INTERPRET O AS CONDITIONAL OF $y | x$... • **NO CI** → RECUR PAST ys

BIDIRECTIONAL RNN

WE WANT TO OUTPUT PREDICTIONS DEPENDING ON THE WHOLE INPUT SEQUENCE (EVEN IN FUTURE) IE SPEECH RECOGNITION / HANDWRITING RECOGNITION, SEQUENCE-TO-SEQUENCE TASKS

- COMBINATION OF FWD AND BWD RNN
- O DEPENDS ON BOTH PAST AND FUTURE
  BUT MOST SENSITIVE AROUND t W/O EXPLICITING WINDOW



- FOR IMAGES, HAVE FOUR NETS. U, D, L, R.

ENCODER-DECODER, SEQUENCE-TO-SEQUENCE ARCHITECTURES

MAP INPUT SEQUENCE TO OUTPUT SEQUENCE OF NOT NEC.LY SAME LENGTH. SPEECH RECOGNITION, TRANSLATION, QUESTION ANSWERING

- CONTEXT: RNN INPUT. → WE WANT TO OBTAIN REPRESENTATION C, VECTOR OR VECTOR SEQUENCE SUMMARIZING THE INPUT.

IDEA: - ENCODER: READER, INPUT RNN → EMITS C AS SIMPLE FCN OF ITS FINAL STATE
- DECODER: WRITER, OUTPUT RNN → CONDITIONED ON C TO EMIT $y = (y_1 \dots y_n)$. LENGTHS CAN VARY ON TRAINING PAIRS. CONDITION EITHER AS STARTING STATE OR EXTRA INPUT

- JOINT TRAINING MAXIMIZES AVG $\log P(y = Y \mid x = X)$.
- h_{ENC}, h_{DEC} MIGHT NOT HAVE SAME DIMENSIONALITY, POSSIBLY COMPLEX, NONLINEAR MAPPING BETWEEN C AND X_{DEC} IE MLP
- ISSUE: WHEN $|C|$ IS TOO SMALL TO SUMMARIZE EFFECTIVELY. MAKE IT VARIABLE LENGTH OR INTRODUCE ATTENTION MECHANISM.
  ↳ ASSOCIATES ELEMENTS OF C TO OUTPUT SEQUENCE ELEMENTS

DEEP RECURRENT NETWORKS

INSOFAR EVERY PARAM BLOCK OF RNN IS SHALLOW. $I \to S$, $S_t \to S_{t+1}$, $S \to O$. SINGLE WEIGHT MATRICES. LET'S ADD DEPTH FOR IE, MODEL DIFFERENT TIMESCALES

- MULTIPLE, HIERARCHICAL HIDDEN LAYERS, EACH UPDATED AT DIFFERENT TIME MULTIPLES. 'PARALLEL'.
- MLP BLOCKS FOR EACH BLOCK. DEEP STATE TRANSITIONS MIGHT HURT. ADD SKIP, DIRECT CONNECTIONS TO KEEP SHORTEST PATHS SHORT.

RECURSIVE NEURAL NETWORKS

COMPUTATIONAL GRAPH IS NOW A DEEP TREE, CAN PROCESS DATA STRUCTURES AS INPUT.
- ADVANTAGE! COMPUTATIONAL DEPTH REDUCED FROM $O(n)$ TO $(\log n)$
- TRICKY HOW TO STRUCTURE THE TREE → POSSIBLY LEARN IT.
- MIGHT NOT USE STD ANN OPS → TENSOR OPS, BILINEAR FORMS

LONG-TERM DEPENDENCIES

EXPLODING / VANISHING GRADIENT ARE A PROBLEM. EVEN IF STABLE, LTD HAVE EXPONENTIALLY SMALLER WEIGHTS VS STD. BECAUSE MANY JACOBIANS ARE MULTIPLIED
↳ IN $O(\lambda^T)$ SPECTRAL RADIUS

- ECHO STATE NETWORKS

AKA RESERVOIR COMPUTING, IDEA: SET WEIGHTS SUCH THAT RECURRENT UNITS DO A GOOD JOB OF CAPTURING HISTORY OF PAST INPUTS • HIDDEN UNITS: TEMPORAL FEATURE RESERVOIR

- ARBITRARY LENGHT INPUT INTO FIXED LENGTH TIME STATE ONTO WHICH LINEAR PREDICTOR. CONVEX IN THE PARAMETERS.

IDEA: DYNAMICAL SYSTEM ASSOCIATED TO RNN HAS TO BE ON THE EDGE OF STABILITY → STATE-TO-STATE TRANSITION FCN JACOBIAN LEADING EIGENVALS ≈ 1
- → BECAUSE EIGENVALUE SPECTRUM OF THE JACOBIANS $J^{(t)} = \frac{\partial s_t}{\partial s_{t-1}}$ → SPECTRAL RADIUS = $\underset{\lambda}{\text{ARGMAX}} |\lambda|$ OF) BECAUSE DYNAMICAL SYSTEMS THEORY
- → MAKE JACOBIANS WEAKLY CONTRACTIVE <1 SO AFTER A WHILE MOST DISTANT PATH IS 'FORGOTTEN'. HOWEVER GOOD RESULT IN PRACTICE WITH 1.2
- → RETAINED INFORMATION IS STABLE, NO VANISHINGS, NO EXPLOSIONS

- MULTIPLE DELAY PATHS

HAVE MULTIPLE, DIFFERENTLY DELAYED, RECURRENT ARCS. → EXPLOSION/VANISHING NOW OCCURS IN $O\left(|\lambda|^{T/d}\right)$ → LONGER DEPENDENCIES

- LEAKY UNITS & TIMESCALE HIERARCHY

'SMOOTH' VARIANT OF MULTIPLE DELAY ARCS ON SELF CONNECTIONS. LINEAR RECURSIVE ARCS + $W \approx 1$ • $s_{t+1,i} = \left(1 - \frac{1}{\tau_i}\right) s_{t,i} + \frac{1}{\tau_i} \sigma\left(b_i + W_i s_t + U_i x_t\right)$ $1 \le \tau \le \infty$

- $\tau = 1$ NO LINEAR SELF RECURRENCE, STD RNN
- $\tau = \infty$ WEIGHTS ARE ALL SAME, SIMPLE AVG OF PAST CONTRIBUTIONS, CAN TURN INTO SUM REMOVING $1/\tau$
- NORMALLY τ EXPONENTIALLY DECAYING WEIGHTS

- HAVE MULTIPLE LEAKY UNITS WITH DIFFERENT TIME SCALES IN THE NET, → τ CAN BE HARDCODED, SAMPLED, OR LEARNED
- STRENGTH OF CONNECTION IS ACTUAL TIMESCALE → TO HAVE DERIVATIVES THROUGH TIME ≈ 1

LONG-SHORT-TERM-MEMORY (LSTM) & OTHER GATED FRIENDS

LEAKY UNITS ALLOW INFO TO BE ACCUMULATED. HOWEVER, WE MIGHT WANT THE NETWORK TO FORGET ITS STATE AT SOME POINT. IE SUBSEQUENCES.
IDEA! LET'S HAVE THE NET LEARN TO DECIDE WHEN TO FORGET

LSTM

CONDITIONING ON THE FORGETTING, LINEAR SELF LOOPS. SELF LOOP WEIGHT IS GATED (CONTROLLED BY ANOTHER HIDDEN UNIT). INTEGRATION TIMESCALE CHANGED DYNAMICALLY
EXTREMELY SUCCESSFUL. FOR LONG-TERM DEPENDENCIES

- **LSTM CELL**

REPLACES NEURON. UNIT. SAME INPUT/OUTPUTS. BUT HAS GATING UNITS (WITH PARAMS) CONTROLLING ITS BEHAVIOR.

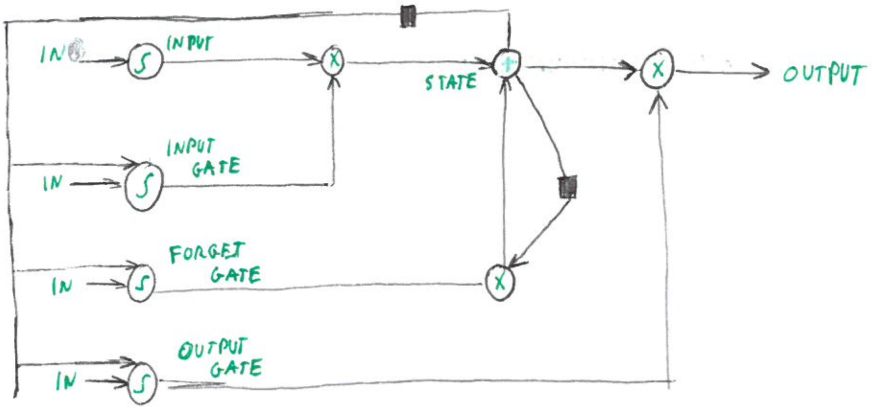
- STATE UNIT $s_{t,i}$ HAS LINEAR SELF-LOOP SIMILAR TO LEAKY UNITS
- FORGET GATE UNIT $h^f_{t,i}$ CONTROLS STATE UNIT LOOP WEIGHT/TIME CONSTANT VIA SIGMOID

$$h^f_{t,i} = \text{SIGM}\left(b^f_i + \sum_j U^f_{i,j} x_{t,j} + \sum_j W^f_{i,j} h_{t,j}\right)$$

b, U, W = FORGET BIASES, IN WEIGHTS, RECURRENT WEIGHT

h_t ↳ HIDDEN LAYER VECTOR, OUTPUT OF ALL LSTM CELLS

- STATE UPDATE: $s_{t+1,i} = \underbrace{h^f_{t,i}}_{\text{SELF-LOOP WEIGHT}} \cdot s_{t,i} + h^e_{t,i}\,\sigma\left(b_i + \sum_j U_{i,j} x_{t,j} + \sum_j W_{i,j} h_{t,j}\right)$
- EXTERNAL INPUT GATE: $h^e_{t,i} = \text{SIGMOID}\left(b^e_i + \sum_j U^e_{i,j} x_{t,j} + \sum_j W^e_{i,j} h_{t,j}\right)$
- OUTPUT GATE $h^o_{t,i} = \text{SIGM}\left(b^o_i + \sum_j U^o_{i,j} x_{t,j} + \sum_j W^o_{i,j} h_{t,j}\right)$
- CELL OUTPUT $h_{t+1,i} = \tanh\left(s_{t+1,i}\right) h^o_{t,i}$



OTHER GATED RNN

OTHER ARCHS TO ALLOW NET TO DYNAMICALLY CONTROL ITS OWN FORGETTING TIMESCALE & RATE?

- **GATED RECURRENT UNITS (GRU)** STATE OF THE ART ENGLISH - FRENCH TRANSLATION. SINGLE GATING UNIT FOR SIMULTANEOUS CONTROL OF FORGETTING FACTOR AND STATE UPDATE. MAKES SENSE IN CONTINUOUS TIME INTERPRETATION. RESET AND UPDATE GATES CAN IGNORE PARTS OF STATE VECTOR, RESET CONTROLS PARTS OF STATE TO USE TO COMPUTE NEXT TARGET STATE. UPDATE GATE ARE CONDITIONAL LEAKY INTEGRATORS, CAN CHOOSE TO COPY IT OR IGNORE IT.
- OTHER VARIANTS IE BY SHARING GATES ACROSS MULTIPLE UNITS. LOCAL/GLOBAL GATES. ETC... BIASES +1 LSTM IS AS STRONG AS ANY OTHER VARIANT SO FAR

EXPLICIT MEMORY

USUAL ANN ARE GREAT AT STORING IMPLICIT, SUBSYMBOLIC, KNOWLEDGE BUT SUCK AT MEMORIZING FACT, SYMBOLIC INFO. SGD
— SGD TAKES MANY ITERATIONS TO STORE STUFF, AND NOT EVEN EXACTLY. IDEA: LET'S ADD WORKING MEMORY

NEURAL TURING MACHINES

LSTM OR GRU - LIKE ADDRESSABLE MEMORY CELLS. NETWORK OUTPUTS STATE CHOOSING WHICH CELL TO READ FROM/WRITE TO

- HARD TO OPTIMIZE FCNS ON EXACT INTEGERS → R/W OPS OVER MANY CELLS SIMULTANEOUSLY. READ: WEIGHTED AVG WRITE: MULTIPLY BY DIFF. AMOUNTS
- COEFFICIENTS FOCUS ON LIMITED NO. OF CELLS → SOFTMAX, DERIVATIVE WEIGHTS → CAN OPTIMIZE WITH SGD
- MEMORY CELLS OFTEN CONTAIN VECTOR BECAUSE BETTER PAYOFF FOR COST OF HAVING THEM. ALSO ALLOW CONTENT-BASED ADDRESSING!! RETRIEVING FROM PATTERN SORTA MATCHING ITS CONTENTS
- STORED INFO CAN BE PROPAGATED FWD IN TIME AND GRADIENTS SAFELY SENT BWD
- SEEMINGLY MORE POWERFUL THAN RNN/LSTM. • ALTERNATIVE: WEIGHTS ARE PROBS, READ 1 CELL ONLY.
- ADDRESS-CHOOSING MECHANISM IS ANALOGOUS TO ATTENTION MECHANISM

BETTER OPTIMIZATION

ALWAYS FOR VANISHING/EXPLODING GRADIENT ISSUE

- **2ND ORDER OPTIMIZATION METHODS** ALLOW DIFFERENT TREATING OF DIFFERENT DIRECTIONS. MANIPULATE GRAD/HESSIAN WE CAN RESCALE STUFF TO MAKE IT STABLE
  → TOO BAD THEY'RE GEARED TOWARDS BATCH PROCESSING

GRADIENT CLIPPING

LANDSCAPES ARE HIGHLY NONCONVEX. EVEN SMALL, DECAYING LR MIGHT FUCK US OVER AND BRING US IN A WORSE PLACE

CLIP DAT GRADIENT:

- CLIP MINIBATCH PARAM GRADIENT ELEMENT-WISE BEFORE PARAM UPDATE
- RANDOM STEP WHEN ABOVE THRESHOLD
- CLIP ~~PARAM~~ GRADIENT NORM BEFORE PARAM UPDATE → STILL IN ORIGINAL GRADIENT DIRECTION

$$\|g\| > v \rightarrow g = \frac{gv}{\|g\|}$$

— INTRODUCES HEURISTIC BIAS IN g ESTIMATION, A USEFUL BIAS

REGULARIZING

FOR VANISHING GRADIENTS. • **IDEA:** LET'S BACKPROP $\nabla_s L$ WHILE MAINTAINING ITS MAGNITUDE → $\nabla_s L$ AS LARGE AS $\nabla_s L \frac{\partial s_t}{\partial s_{t-1}}$

$$- \Omega = \sum_t \left(\frac{\| \nabla_{s_t} L \frac{\partial s_t}{\partial s_{t-1}}}{\| \nabla_{s_t} L} \right)^2$$

AND APPROXIMATE $\nabla_s L$ TO CONSTANTS • **THIS + NORM CLIPPING:** SIZABLE INCREASE TO SUCCESSFUL DEPENDENCY SPAN LEARNED

STATE AS MULTIPLE TIMESCALES

MODEL HIERARCHIC ARCHITECTURE. DIFFERENT HIDDEN LAYERS, DIFFERENT TIMESCALES. LEAKY UNITS WITH DIFFERENT τ, EXPLICIT SKIPPED UPDATES...