1) Ghazal Kaviani

Zhaoxin Li

2.1)

$$f(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{w}^T\mathbf{x} + b \geq 0 \\ 0 & \text{if } \mathbf{w}^T\mathbf{x} + b < 0 \end{cases}$$

Pair	x_1	x_2	$f_{\text{AND}}(\mathbf{x})$
1	0	0	0
2	0	1	0
3	1	0	0
4	1	1	1

Find w & b

Try $w = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $b = -2$

Pair 1: $w^T x + b = \begin{bmatrix} 1 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \end{bmatrix} - 2 = 0 - 2 = 0 \Rightarrow f(x) = 0$ ✓

Pair 2: $\begin{bmatrix} 1 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} - 2 = 1 - 2 = -1 \Rightarrow f(x) = 0$ ✓

Pair 3: $\begin{bmatrix} 1 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} - 2 = -1 \Rightarrow f(x) = 0$ ✓

Pair 4: $\begin{bmatrix} 1 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix} - 2 = 2 - 2 = 0 \Rightarrow f(x) = 1$ ✓

By inspection, $\boxed{w_{\text{AND}} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \text{ & } b_{\text{AND}} = -2}$ work

Now, for OR:

Pair	x_1	x_2	$f_{\text{OR}}(\mathbf{x})$
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	1

Try $w_{\text{OR}} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $b_{\text{OR}} = -0.5$

Pair 1: $w^T x + b = \begin{bmatrix} 1 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \end{bmatrix} - 0.5 = -0.5 \Rightarrow f(x) = 0$ ✓

Pair 2: $\begin{bmatrix} 1 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} - 0.5 = 1 - 0.5 = 0.5 \Rightarrow f(x) = 1$ ✓

2.1) cont'd

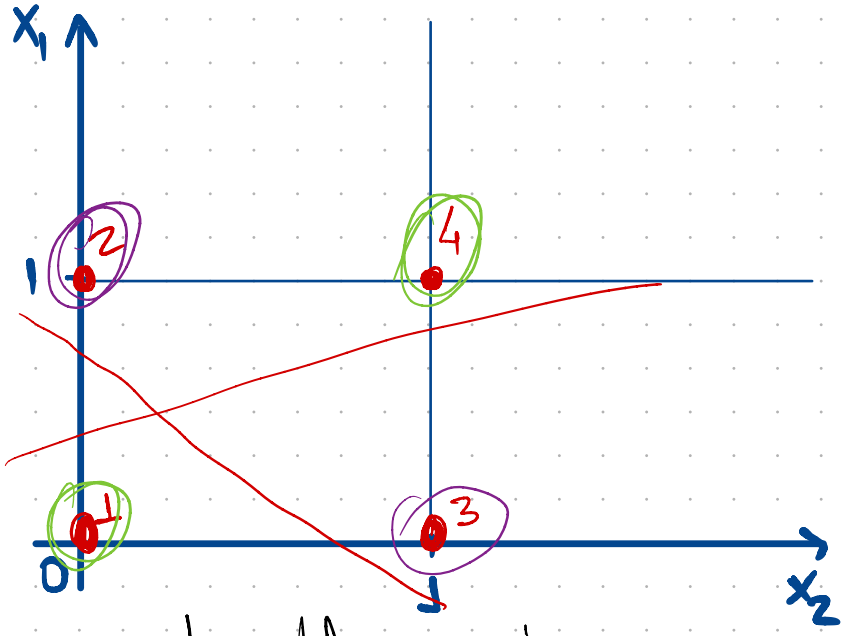
Pair 3: $\begin{bmatrix}1 & 1\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix} - 0.5 = 1 - 0.5 = 0.5 \Rightarrow f(x) = 1$ ✓

Pair 4: $\begin{bmatrix}1 & 1\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix} - 0.5 = 2 - 0.5 = 1.5 \Rightarrow f(x) = 1$ ✓

Similarly, by inspection, $\boxed{W_{02} = \begin{bmatrix}1\\1\end{bmatrix} \text{ & } b_{02} = -0.5}$ work

2.2)

Pair	x_1	x_2	$f_{\text{XOR}}(x)$
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	0



By inspection in the 2D plane, there is no possible line to separate the pairs corresponding to the same f_{xor} value $\Rightarrow$ the XOR function isn't linearly separable

Formally, we try to find rules for a line using w & b (Assume it is separable — proof by contrad):

Pair 1 < Pair 2 $\quad (f_{xor}(\text{Pair 1}) = 0 \,;\, f_{xor}(\text{Pair 2}) = 1)$

$$w^T \begin{bmatrix} 0 \\ 0 \end{bmatrix} + b < w^T \begin{bmatrix} 0 \\ 1 \end{bmatrix} + b \Rightarrow b < w_1 \cdot 0 + w_2 \cdot 1 + b$$

$w_2 > 0$ (I) 1ST rule

Pair 4 < Pair 3 $\quad (f_{xor}(\text{Pair 4}) = 0 \,;\, f_{xor}(\text{Pair 3}) = 1$

$$w^T \begin{bmatrix} 1 \\ 1 \end{bmatrix} + b < w^T \begin{bmatrix} 1 \\ 0 \end{bmatrix} + b \Rightarrow w_1 + w_2 < w_1$$

$w_2 < 0$ (II) Rules (I) and (II) lead to a contradiction ■

3.3) Objective: Design a LSTM that maps x to y in the XOR operation.

Usual notation (x for input, y for output), and taking advantage of the hint:

$(x \wedge \bar{y}) \vee (\bar{x} \wedge y)$ = XOR operation → Notation $_t$ because it's a recursive scheme

$$y_t = (x_t \wedge \bar{y}_{t-1}) \vee (\bar{x}_t \wedge y_{t-1})$$ Remind gate-to-algebra

$$= x_t * \bar{y}_{t-1} + \bar{x}_t * y_{t-1} \quad (I)$$

Mapping and matching with eqn (5) from the PS:

(I) & (5): $C_t = x_t * y_{t-1} + \bar{x}_t * y_{t-1} = f_t * C_{t-1} + i_t * \tilde{C}_t$

Yielding: $C_t = y_t$ = XOR output
$f_t = \bar{x}_t$, $i_t = x_t$ } (II)

As per RNN's nature, there is also the hidden state h_t, which in this case maps to C_t (and y_t consequently). → $C_t = y_t = h_t$

Now we can match (II) and eqn (2) →

3.3 cont'd) eqn (2): $f_t = \sigma(W_f \cdot [h_{t-1}\ x_t] + b_f)$

mapping & matching with (I) and (II):

$f_t = \overline{x}_t = \sigma(W_f \cdot [h_{t-1}\ x_t] + b_f)$

$x_t = 0 \Rightarrow \overline{x}_t = 1 = f_t$
$x_t = 1 \Rightarrow \overline{x}_t = 0 = f_t$ } (III)

Inspecting:

$$\begin{bmatrix} W_{f1} \\ W_{f2} \end{bmatrix} [h_{t1}\ x_t] + b_f = W_{f1} h_{t-1} + W_{f2} x_t + b_f = f_t$$

didn't even have to touch the σ — $W_{f1} h_{t-1} \to 0$; $W_{f2} x_t \to -1$; $b_f \to 1$

Setting $W_{f1} = 0 \Rightarrow W_{f2}$ has to be opposite to b_f so the rule (III) holds

$\therefore$ $\boxed{W_f = \begin{bmatrix} 0 \\ -1 \end{bmatrix},\ b_f = 1}$

Similar matching, but with eqn. (3)

$i_t = x_t = \sigma(W_i [h_{t-1}\ x_t] + b_i)$

$x_t = i_t = 0$ & $x_t = i_t = 1$

inspecting: $\begin{bmatrix} W_{i1} \\ W_{i2} \end{bmatrix} [h_{t-1}\ x_t] + b_i = W_{i1} h_{t-1} + W_{i2} x_t + b_i$ →

3.3 cont'd) $W_{i1}h_{t-1} + W_{i2}x_t + b_i = x_t$

(W_{i1}: 0, W_{i2}: 1, b_i: 0)

Also by inspection:

$$W_i = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad b_i = 0$$

Again no need to mess with σ

Continuing the same process for eqn (4):

$$\bar{C}_t = \tanh(W_c[h_{t-1}\ x_t] + b_c)$$

$C_t = y_t = h_t \Rightarrow \bar{C}_t = \bar{y}_t = \bar{h}_t$ (Again try to do everything inside the activation func):

$W_{c1}h_{t-1} + W_{c2}x_t + b_c = \bar{h}_t \rightarrow \tanh = 0 \Leftrightarrow x = 0$

$\tanh = 1 \Leftrightarrow x > 0$

potentially degenerate so won't zero out b_c

Try $b_c = 1$

$\bar{h}_t = \tanh(-h_{t-1} + 1) \rightarrow$

- $h_{t-1} = 0 \Rightarrow h_t = 1$
- $h_{t-1} = 1 \Rightarrow h_t = 0$ ✓

By inspection:

$$W_c = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \quad b_c = 1$$

3.3 cont'd) Last to match and inspect is eqn. (6) & (7)

$h_t = o_t * \tanh(C_t)$, $o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$

$h_t = y_t = C_t$

$\tanh(C_t)$ will dominate

$\tanh(C_t = 1) = 1 = h_t$
$\tanh(C_t = 0) = 0 = h_t$
$\Rightarrow$ leave $o_t = 1$

$\sigma(W_o \cdot [h_{t-1}, x_t] + b_o) = 1$

$W_{o1} h_{t-1} + W_{o2} x_t + b_o = 1$

Isolate since we want o_t fixed at 1

$\therefore$ $W_o = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $b = 1$

Copying all other learned weights to the same pg for simplicity:

$W_f = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$, $b_f = 1$

$W_i = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $b_i = 0$

$W_C = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$, $b_C = 1$

DESIGN COMPLETE!

3.4) Want to prove that B_t (highest scoring beam) scores worse than or equal to the $best_{<t}$

Let us define $f(\cdot)$ as the score

Thus far we have $f(best_{<t})$ as the highest

By the claim, $f(best_{<t}) \geq$ any other $f(\cdot)$

Assume B_t (y'), which happens after $best_{<t}$

i.e, t

$$f(best_{<t}) \geq f(B_t)$$

Assume another beam at another time later than t (t^+)

By the claim: $f(B_{t^+}) \leq f(B_t)$

Scores are upperbounded by nature

Hence $f(B_{t^+}) \leq f(best_{<t})$ ■

3.5) Eqn (14): $h_t = W^T h_{t-1}$ is recurring. Going backwards, we can see the first hidden state established relationship: $h_1 = W^T h_0$

Carrying forward: $h_t = W^T \cdot W^T \cdot \dots h_0$

$$h_t = (W^T)^t h_0$$

We can observe that anything that is contained in W will propagate in a potentially huge scale downstream because it is to the power t, i.e., to the power of the timesteps

Intuitively, we know that this means that big numbers can lead to overflow (exploding gradients) and small can lead to vanishing.

We can see how this relates to the eigenvalues by performing eigendecomposition / diagonalization (common, not SVD) of W:

$W = V \Lambda V^{-1}$ (Λ is a diagonal matrix containing W's eigenvalues in the diagonal)

$$h_t = (V \Lambda V^{-1})^T{}^t h_0$$

$$= ((V^{-1})^T \Lambda^T V^T)^t h_0 = (V^{-T} \Lambda^T V^T)^t$$

expanding → continued →

3.5 cont'd) expanding:

$$h_t = [(V^{-T}\Lambda^T V^T)\cdot(V^{-T}\Lambda^T V^T)\cdots(V^{-T}\Lambda^T V^T)]$$

$$V^T V^{-T} = I$$

$$h_t = V^{-T}\Lambda^t V \longrightarrow$$ eigenvectors cancel out eventually, but we are still left with Λ^t

Confirming the intuition: All the "essential information" is "stored" and passed via a matrix's eigenvalues, which are the numbers that are to the power t.

More formally, there are 3 cases:

(Let us refer to the eigenvalues that constitute Λ as σ_i)

1. σ_i are too small.
   $t \to \infty \Rightarrow \sigma_i \to 0$ (vanishing gradients)

2. σ_i are too big (not necessarily positive).
   $t \to \infty \Rightarrow \sigma_i \to \infty$ (exploding gradients)

3. Not 1 and Not 2, which is where we want to stay.

CS 4644/7643: Deep Learning
Fall 2022
HW3 Solutions

Arthur Scaquetti do Nascimento

March 12, 2024

4.7)

- Key contributions: This work proposes a technique for pretraining sequenc-to-sequence models with denoising objectives for learning general-purpose representations, which was thought for NLP, but is a cornerstone to transfer to other modalities. It explores different techniques for denoising, is by design able to handle different NL processing tasks, and was the benchmark to be beaten when it came out.
- Strengths:
  - It can handle different tasks without task-specific modifications, which leads one to infer that its generalization capabilities are very solid.
  - The method proposed is shown to be very robust by nature, i.e., the denoising foundations of it make it so that it is able to handle corrupted data,
- Weaknesses: As expected, BART is extremely computational complex. On top of that, I didn't find an effort into pinpointing the bottleneck, what aspect of BART contributes more to its computation complexity. Also, we still don't find interpretability or explainability investigated.

4.8) Personal takeaways:
While reading, it bothered my a lot that the latest research in this large models filed oversees the causes of computation complexity. Yes, there is an obvious direct correlation between the number of parameters of a model and its complexity, but I am starting to wonder if there is something more to it than that.
As in what methods, operations or techniques take more compute? If it is the denoising process, which part? I have enver seen an ablation study like this myself. Moreover, it get me thinking about the first PS: up to what point it is the architecture's fault? I have no idea how to answer that, but it's something that got me thinking.