

Deep Recurrent Neural Networks for Prostate Cancer Detection: Analysis of Temporal Enhanced Ultrasound

- Shekoofeh Aziz et al.

Mitodru Niyogi

Interdisciplinary Center of Scientific Computing, Heidelberg University

June 25, 2020





- 1 Introduction
- 2 Background
- 3 Dataset
- 4 Proposed Model
- 5 Experiments
- 6 Results and Analysis
- 7 Conclusion
- 8 Reference

Objective:

- Develop a deep learning model to discriminate cancer and benign prostate regions in TeUS data.
- **Why deep learning?**
 - It automatically learns a high-level latent feature representation from data without handcrafted features.
- **Common approaches:**
 - Ultrasound Imaging for tissue characterization, Elastography, Doppler Imaging, fusion of transrectal ultrasound (TRUS) with multi-parametric MRI (mp-MRI) have been used for PCa detection.
- **Modern approaches:**
 - Machine learning framework such as random forests, support vector machines, Bayesian classifiers, Hidden Markov Models have been used to extract information from the backscattered ultrasound data obtained.

Prostate Cancer insights

- The American Cancer Society estimates 191,930 new cases to be diagnosed and 33,330 deaths in 2020 ¹ in the United States. Early stage PCa detection followed by treatment results in a five-year survival rate of above 95% [3].
- About 1 man in 9 will be diagnosed with prostate cancer during his lifetime.
- Prostate cancer is the second leading cause of cancer death in American men, behind only lung cancer.
- About 1 man in 41 will die of prostate cancer.

¹<https://www.cancer.org/cancer/prostate-cancer/about/key-statistics.html>     5/27

Recurrent Neural Network (RNN)

- RNNs process sequential data points through a recurrent hidden state whose activation at each step depends on that of a previous step. Generally, given sequence data $x = (x_1, \dots, x_T)$, an RNN updates its recurrent hidden state h_t by

$$h_t = \begin{cases} 0, & \text{if } t = 0 \\ \varphi(h_{t-1}, x_t), & \text{otherwise} \end{cases} \quad (1)$$

where x_t and h_t are input and the recurrent hidden state at time step t , and $\varphi(\cdot)$ represents the nonlinear activation function of a hidden layer, such as a sigmoid or hyperbolic tangent

- In vanilla RNN, the update rule of the recurrent hidden state

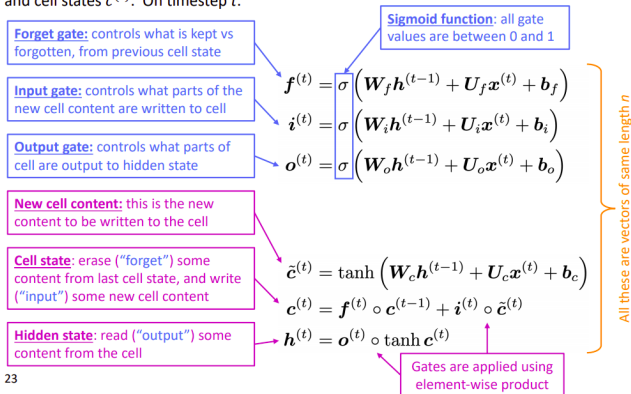
$$h_t = \varphi(Wx_t + Uh_{t-1}), \quad (2)$$

$$h_t = \Phi(W_{ih}x_t + W_{hh}h_{t-1} + b_h), \quad (3)$$

where $t = 1$ to T , W_{ih} denotes the input-hidden weight vector, W_{hh} represents the weight matrix of the hidden layer, and b_h is the hidden layer bias vector

Long Short-Term Memory network (LSTM)

We have a sequence of inputs $x^{(t)}$, and we will compute a sequence of hidden states $h^{(t)}$ and cell states $c^{(t)}$. On timestep t :



23

Figure: Fig taken from <http://web.stanford.edu/class/cs224n/slides/cs224n-2020-lecture07-fancy-rnn.pdf>

Gated Recurrent Unit (GRU)

- Proposed by Cho et al. in 2014 as a simpler alternative to the LSTM.
- On each timestep t we have input $\mathbf{x}^{(t)}$ and hidden state $\mathbf{h}^{(t)}$ (no cell state).

Update gate: controls what parts of hidden state are updated vs preserved

$$\mathbf{u}^{(t)} = \sigma \left(\mathbf{W}_u \mathbf{h}^{(t-1)} + \mathbf{U}_u \mathbf{x}^{(t)} + \mathbf{b}_u \right)$$

Reset gate: controls what parts of previous hidden state are used to compute new content

$$\mathbf{r}^{(t)} = \sigma \left(\mathbf{W}_r \mathbf{h}^{(t-1)} + \mathbf{U}_r \mathbf{x}^{(t)} + \mathbf{b}_r \right)$$

New hidden state content: reset gate selects useful parts of prev hidden state. Use this and current input to compute new hidden content.

$$\tilde{\mathbf{h}}^{(t)} = \tanh \left(\mathbf{W}_h (\mathbf{r}^{(t)} \circ \mathbf{h}^{(t-1)}) + \mathbf{U}_h \mathbf{x}^{(t)} + \mathbf{b}_h \right)$$

$$\mathbf{h}^{(t)} = (1 - \mathbf{u}^{(t)}) \circ \mathbf{h}^{(t-1)} + \mathbf{u}^{(t)} \circ \tilde{\mathbf{h}}^{(t)}$$

Hidden state: update gate simultaneously controls what is kept from previous hidden state, and what is updated to new hidden state content

How does this solve vanishing gradient?

Like LSTM, GRU makes it easier to retain info long-term (e.g. by setting update gate to 0)

Figure: Fig taken from <http://web.stanford.edu/class/cs224n/slides/cs224n-2020-lecture07-fancy-rnn.pdf>

Dataset

- TeUS data was acquired from 157 subjects during fusion prostate biopsy.
- In dataset, there are 83 biopsy cancerous cores with GS 3+3 or higher, 31 cancerous cores with $GS \geq 4+3$. The remaining 172 cores are non-cancerous and include benign or fibromuscular tissue.
- Training data: consists of 84 cores with the following histopathology label distribution: benign: 42 cores; 2 cores of GS 3+3; 14 cores of GS 3+4; 3 cores of GS 4+3; 18 cores of GS 4+4; and, 5 cores of GS 4+5.
- Test data: consists of 171 cores, where 130 cores are labeled as benign, 29 cores with $GS \leq 3+4$, and 12 cores with $GS \geq 4+3$.
- Train-Validation split: 80-20%

Data Preprocessing and augmentation

- For each biopsy target, target region of 2 mm × 10 mm area around the target location has been considered.
- The target region is divided to 80 ROIs of size 0.5 mm × 0.5 mm.
- Sliding window of size 0.5 mm × 0.5 mm approach is used for the data augmentation.
- Number of training samples ($N = |D_{train}| = 129,024$).

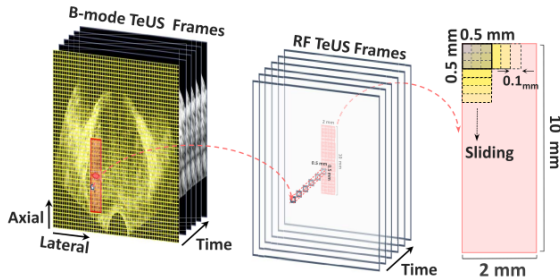


Figure: Preprocessing and ROI selection. Fig taken from [1]

Proposed Discriminative Method

- Let $D = (x^{(i)}, y^{(i)})_{i=1}^{|D|}$ represent a collection of all labeled ROIs, where $x^{(i)}$ is the i^{th} TeUS sequence and $y^{(i)}$ indicates the corresponding label.
- $x^{(i)} = (x_1^{(i)}, \dots, x_T^{(i)})$, is composed of signal-amplitude values $x_t^{(i)}$ for each time step, t , and is labeled as $y^i \in \{0, 1\}$, where zero and one indicate benign and cancer outcome,
- Training Objective**
 - The model learns a distribution over classes $P(y|x_1, \dots, x_T)$ given a time-series sequence x_1, \dots, x_T .
 - The final hidden state generates the posterior probability for the given sequence:

$$z(i) = w_s^T h + b_s; \quad (4)$$

$$\bar{y}^{(i)} = P(y_j^{(i)}|x) = S(z_j^{(i)}) = \frac{\exp^{z_j^{(i)}}}{\exp^{z_0^{(i)}} + \exp^{z_1^{(i)}}}, j \in \{0, 1\} \quad (5)$$

where S is the softmax function, which in the binary classification case is equivalent to the logistic function, and $\bar{y}^{(i)}$ indicates the predicted label.

Optimization Criterion: The optimization criterion is to minimize the negative log-likelihood of the loss function which is the binary cross-entropy between $y^{(i)}$ and $\bar{y}^{(i)}$ over all training samples,

$$L(\bar{y}, y) = -\frac{1}{N} \sum_{i=1}^N [y^{(i)} \log \bar{y}^{(i)} + (1 - y^{(i)}) \log(1 - \bar{y}^{(i)})], \quad (6)$$

where $N = |D_{train}|$

Cancer Classification:

- The probability of a given core being cancerous: $P_C = \frac{\sum_i^{|C|} I(\bar{y}^{(i)}=1)}{|C|}$
- Cancerous core, when $P_C \geq 0.5$.

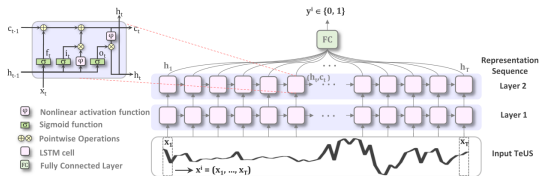


Figure: Proposed method. Fig taken from [1]

Hyperparameter Selection

- Grid search approach is used to optimize the hyper-parameters. It is an exhaustive search through a prespecified subset of the hyperparameter space of the learning algorithm.
- Both regularization (L2 regularization) and dropout have been used to reduce the over-fitting.
- Grid search hyperparameters: the number of RNN hidden layers, $n_h \in \{1, 2\}$, batch size, $b_s \in \{64, 128\}$, and initial learning rate, $lr \in \{0.01, 0.0001\}$ with three different optimization algorithms, SGD, RMSprop and Adam.
- All models are trained with the same number of iterations and training is stopped after 100 epochs.
- Monitoring of the validation loss and if no improvement is observed over 10 epochs, the learning rate is reduced by $lr_{new} = lr \times factor$, where $factor = 0.9$.

Model Training and Evaluation

- Early stopping if loss does not decrease or increases after 10 epochs.
- Sensitivity, specificity, and accuracy in detecting cancerous tissue samples in the test data, D_{test} .
- **Sensitivity** or recall is defined as the percentage of cancerous cores that are correctly identified, while **specificity** is the proportion of non-cancerous cores that are correctly classified.
- The overall performance of the models are reported using AUC.
- The AUC curve depicts a relative trade-off between sensitivity and specificity. The maximum value for AUC is 1, where higher values indicate better classification performance.

Network Analysis: Ablation study

- Examine the LSTM gates to better understand the temporal information in TeUS.

Algorithm 1 Examination of the LSTM Gates

Input: Trained model parameters “ $\Theta = \{\mathcal{W}, \mathcal{B}\}$ ”, input data “ \mathbf{X} ”, number of time-steps “ T ”, number of input sequence “ N ”.

Output: States activation “ \mathcal{S} ”, gates activation “ \mathcal{G} ”

Initialization: Set the state of each cell “ $inStates$ ” to zero.

```

1: for  $i = 0$  to  $N$  do
2:   for  $t = 0$  to  $T$  do
3:      $x \leftarrow \mathbf{X}(i, :)$ 
4:      $\{\mathcal{S}(i, t), \mathcal{G}(i, t)\} \leftarrow \text{STEP}(x, inStates(i, t), \mathcal{W}, \mathcal{B})$ 
5:      $inState(i, t) \leftarrow \mathcal{S}(i, t)$ 
6:   end for
7: end for
8: return  $\mathcal{S}, \mathcal{G}$ 

```

Algorithm 2 Recurrent Step Function of the LSTM

Input: Trained model parameters “ $\Theta = \{\mathcal{W}, \mathcal{B}\}$ ”, input sequence “ \mathbf{x} ”, input states of time step $(t - 1)$ “ \mathcal{S} ”.

Output: States activation of the current time step (t) “ \mathcal{S}_t ”, gates activation of the current time step (t) “ \mathcal{G}_t ”

- 1: **procedure** STEP($x, \mathcal{S}_{t-1}, \mathcal{W}, \mathcal{B}$)
 - 2: $W_{oi}, W_{oh}, W_{oc}, W_{ci}, W_{ch}, W_{ix}, W_{ih}, W_{fx}, W_{fh}, W_{fc} \leftarrow \mathcal{W}$
 - 3: $b_o, b_c, b_i, b_f \leftarrow \mathcal{B}$
 - 4: $h_{t-1}, c_{t-1} \leftarrow \mathcal{S}_{t-1}$
 - 5: $i_t \leftarrow \sigma(W_{ix}x + W_{ih}h_{t-1} + W_{ic}c_{t-1} + b_i)$
 - 6: $f_t \leftarrow \sigma(W_{fx}x + W_{fh}h_{t-1} + W_{fc}c_{t-1} + b_f)$
 - 7: $o_t \leftarrow \sigma(W_{oi}x + W_{oh}h_{t-1} + W_{oc}c_{t-1} + b_o)$
 - 8: $\bar{c} \leftarrow \phi(W_{ci}x + W_{ch}h_{t-1} + b_c)$
 - 9: $c_t \leftarrow i_t \odot \bar{c} + f_t \odot c_{t-1}$
 - 10: $h_t \leftarrow o_t \phi(c_t)$
 - 11: $\mathcal{S}_t \leftarrow \{h_t, c_t\}$
 - 12: $\mathcal{G}_t \leftarrow \{i_t, f_t, o_t, c_t\}$
 - 13: **return** $\mathcal{S}_t, \mathcal{G}_t$
-

Model Selection:

- RMSprop substantially outperforms SGD optimization for all of the RNN cell types.
- RMSprop and Adam optimizers have similar performance for GRU and LSTM cells.
- RMSprop leads to a better performance on the dataset.
- **Optimized models:** $b_s = 128$, $lr = 0.0001$ (vanilla RNN); $b_s = 64$, $lr = 0.0001$ (LSTM); $b_s = 128$, $lr = 0.01$ (GRU).
- For all models, dropout rate, i.e, $d_r = 0.2$ and $l_{reg} = 0.0001$ generate the lowest loss and the highest accuracy for both training and validation datasets.
- All models converge after 65 ± 7 epochs, and GRU and LSTM cells outperform vanilla RNN cells in terms of accuracy.
- GRU cells has a steeper learning curve and converges faster than the network with LSTM cells.

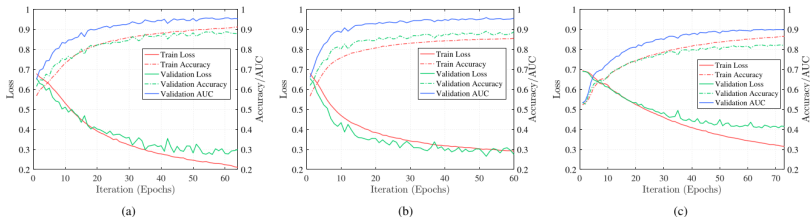


Figure: Learning curves of different RNN cells using the optimum hyper-parameters in our search space. All of the models use the RMSprop optimizer and converge after 65 ± 7 epochs. (a) LSTM. (b) GRU. (c) Vanilla RNN. Fig taken from [1]

Model Performance

TABLE I

MODEL PERFORMANCE FOR CLASSIFICATION OF CORES
IN THE TEST DATA (N = 171)

Method	Specificity	Sensitivity	Accuracy	AUC
LSTM	0.98	0.76	0.93	0.96
GRU	0.95	0.70	0.86	0.92
Vanilla RNN	0.72	0.69	0.75	0.76
Spectral [17]	0.73	0.63	0.78	0.76

MODEL PERFORMANCE FOR CLASSIFICATION OF CORES IN THE TEST DATA FOR DIFFERENT MR SUSPICIOUS LEVELS.
N INDICATES NUMBER OF CORES IN EACH GROUP

MR suspicious levels	Moderate MR suspicious level (N = 115)				High MR suspicious level (N = 20)			
	Specificity	Sensitivity	Accuracy	AUC	Specificity	Sensitivity	Accuracy	AUC
LSTM	0.98	0.78	0.95	0.97	0.86	0.92	0.90	0.97
GRU	0.95	0.70	0.89	0.91	0.80	1.00	0.90	0.92
Vanilla RNN	0.82	0.70	0.82	0.73	0.85	0.67	0.70	0.68
Spectral [17]	0.70	0.78	0.80	0.80	0.83	0.90	0.85	0.95

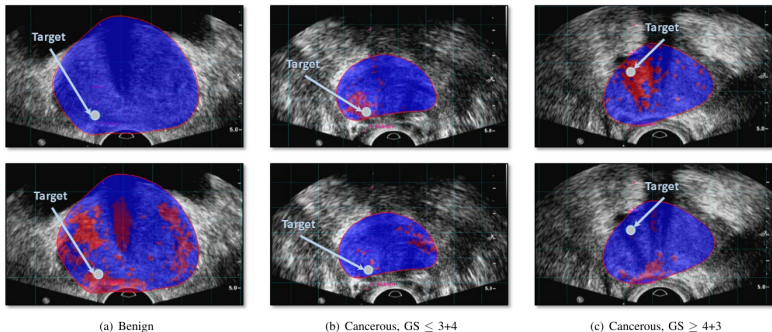


Figure: Cancer likelihood maps overlaid on B-mode ultrasound images, along the projected needle path in the TeUS data, and centered on the target. Red indicates predicted labels as cancer, and blue indicates predicted benign regions. The boundary of the segmented prostate in MRI is overlaid on TRUS data. The arrow points to the target location. The top row shows the result of LSTM and the bottom row shows the result of spectral analysis [2] for benign targets (a), and cancer targets (b) and (c). Fig taken from [1].

Network Analysis

- **What contributes the most to distinguish between benign and cancerous cells?**

The difference map between the final activation of the network (h_t at $t = 100$) for TeUS data from benign and cancerous samples have been generated and top 20 cells with the highest activation difference has been chosen.

- The most discriminative features for distinguishing cancerous and benign tissue samples are captured within the first half of TeUS sequence.

Effect of input sequence length on performance

- The higher the length of input TeUS sequence, the performance of the models increase.
- However, for TeUS sequence length more than 50, the improvement saturates.

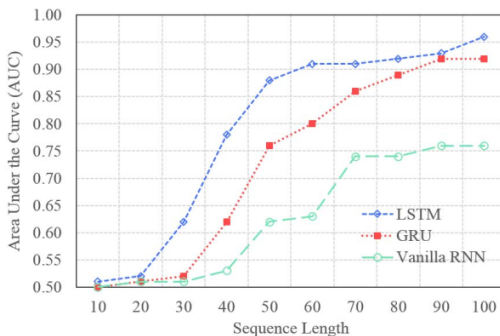


Figure: Sequence length effect. Fig taken from [1]




Conclusion

- The authors achieved an accuracy of 93% by using LSTM based model for prostate cancer detection.
- Successfully conducted statistical significance t-tests with confidence level of more than 95% improvement in accuracy over spectral analysis approach [2] proposed by authors previously .
- Hyperparameter tuning was performed in order to tune the model performance from over-fitting.
- Implementation details were shared along with training time.

Criticism

- No plots or evaluation of LSTM gate cells for visualizing the activations of cells reported.
- Standard RNNs networks background could have been skipped and repetition of statements.
- Not so strong baselines.

Reference

-  Azizi et al. "Deep Recurrent Neural Networks for Prostate Cancer Detection: Analysis of Temporal Enhanced Ultrasound". *IEEE Transactions On Medical Imaging*, vol. 37, no. 12, December 2018
-  S. Azizi, F. Imani, A. Tahmasebi, B. Wood, P. Mousavi, and P. Abolmaesumi, "Detection of prostate cancer using temporal sequences of ultrasound data: A large clinical feasibility study," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 11, no. 6, pp. 947956, 2016.
-  J. O. Barentsz et al., "ESUR prostate MR guidelines 2012," *Eur. Radiol.*, vol. 22, no. 4, pp. 746757, Apr. 2012.

**Thank you very much for your attention.
Do you have any question?**