Multitask Learning and Benchmarking with Clinical Time Series Data

Xiao Yan
2018.11.02

Challenging

- data is irregular in patient level but regular at local episode level
- time gap between two visits in random
- data for patient varies greatly in length
- absence of universally accepted benchmark

PROBLEM DEFINITION

- Mortality:

  label indicates whether the patient died before hospital discharge

- Decompensation:

  the target label indicates whether the patient will die within the next 24 hours

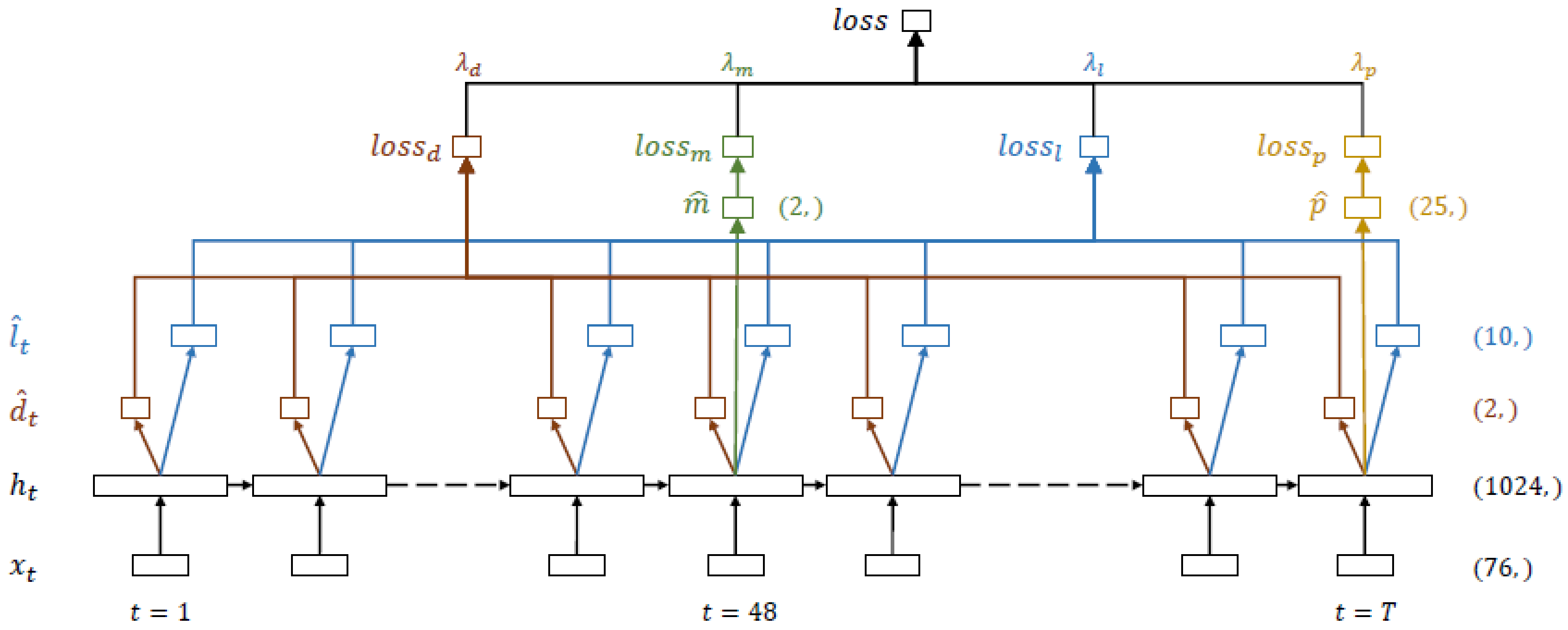
- length of stay:

  Divide the range of value into ten buckets (0-1,1-2,…,7-8,8-14,14+)

- Phenotype:

  25 conditions(12 critical conditions, 8 chronic conditions, 5 “mixed” conditions)

METHOD



Multitask Learning and Benchmarking with Clinical Time Series Data

METHOD

stay of length T hours, observations $\{x_t\}_{t\geq 1}$, x_t is vector of hour

mortality: $m \in \{0,1\}$ is single label indicating whether the patient die

decompensation: $\{d_t\}_{t\geq 1}^{T}$ where $d_t \in \{0,1\}$ is binary labels for each hour

length of stay: $\{l_t\}_{t\geq 1}^{T}$ where $l_t \in R$ is value number at each time step

Phenotyping: $p_{1:K} \in \{0,1\}^K$ is a vector of K binary phenotype labels

$$loss_d = \frac{1}{T}\sum_{t=1}^{T} \mathrm{CE}(d_t, \widehat{d}_t) \qquad loss_p = \frac{1}{K}\sum_{i=1}^{K} \mathrm{CE}(p_k, \widehat{p}_k) \qquad loss_l = \frac{1}{T}\sum_{t=1}^{T} \mathrm{MCE}(l_{tk}, \widehat{l}_{tk}) \qquad loss_m = \mathrm{CE}(m, \widehat{m}) \qquad loss_\ell = \frac{1}{T}\sum_{t=1}^{T}\left(\ell_t - \widehat{\ell}_t\right)^2$$

$$loss_{mt} = \lambda_d \cdot loss_d + \lambda_l \cdot loss_l + \lambda_m \cdot loss_m + \lambda_p \cdot loss_p$$

Multitask Learning and Benchmarking with Clinical Time Series Data

RESULT

Model	AUROC	AUPRC	min(Se, $^+$P)
Logistic regression	0.8442	0.4717	0.4693
LSTM	0.8540	0.5164	0.4905
Multitask LSTM	0.8550	0.4926	0.4745
Multitask LSTM*	0.8625	0.5169	0.4987

mortality

Model	AUROC	AUPRC	min(Se, $^+$P)
Logistic regression	0.8704	0.2132	0.2688
LSTM	0.8946	0.2980	0.3438
Multitask LSTM	0.9004	0.3192	0.3484
Multitask LSTM**	0.9119	0.3322	0.3593

decompensation

Model	Kappa	MSE	MAPE
Logistic regression	0.4021	63385	573.5
LSTM	0.4266	42165	235.9
Multitask LSTM	0.4258	42131	188.5

Length of stay

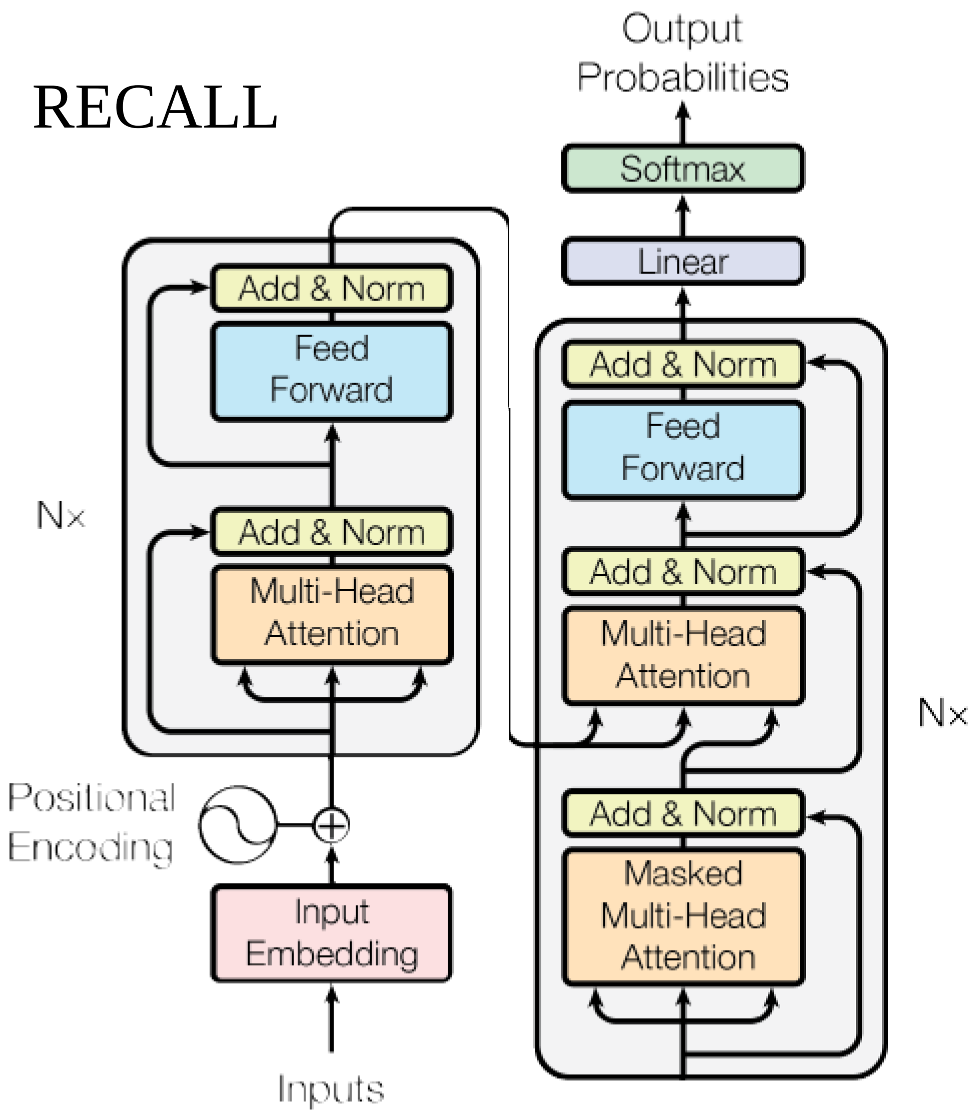
Model	micro AUC	macro AUC	weighted AUC
Logistic regression	0.8007	0.7408	0.7320
1-layer LSTM	0.8206	0.7701	0.7573
2-layer LSTM	0.8213	0.7707	0.7587
Multitask LSTM	0.8174	0.7661	0.7533

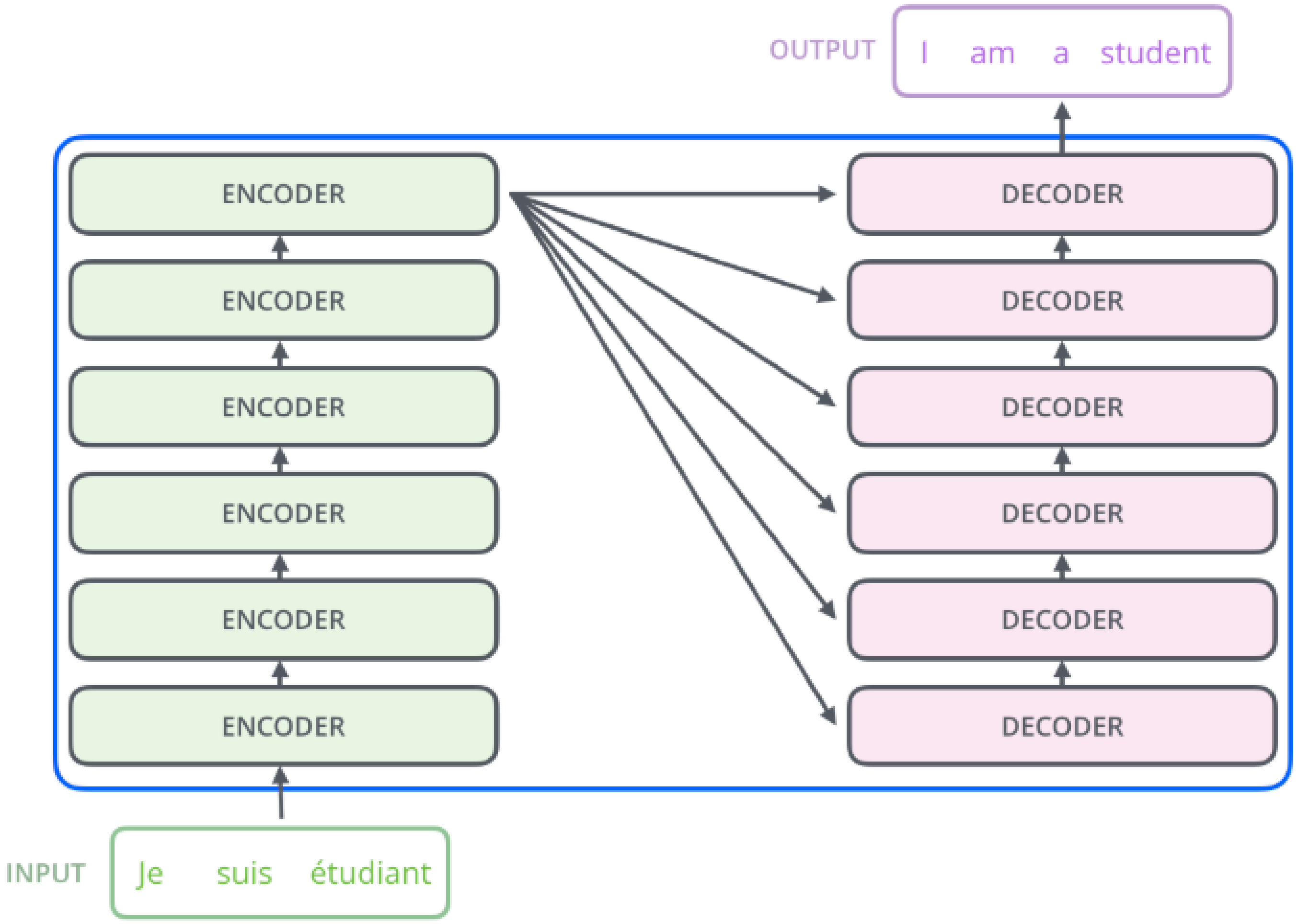
phenotyping

Attend and Diagnose:

Clinical Time Series Analysis Using Attention Models

RECALL



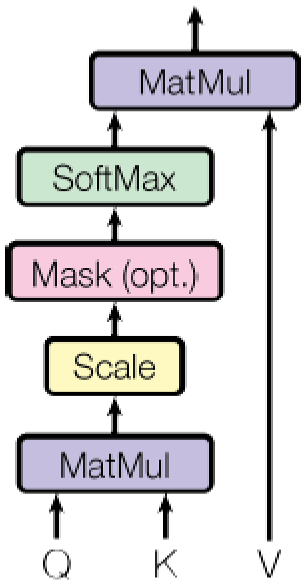


Attend and Diagnose: Clinical Time Series Analysis Using Attention Models

Ref : http://jalammar.github.io/illustrated-transformer/

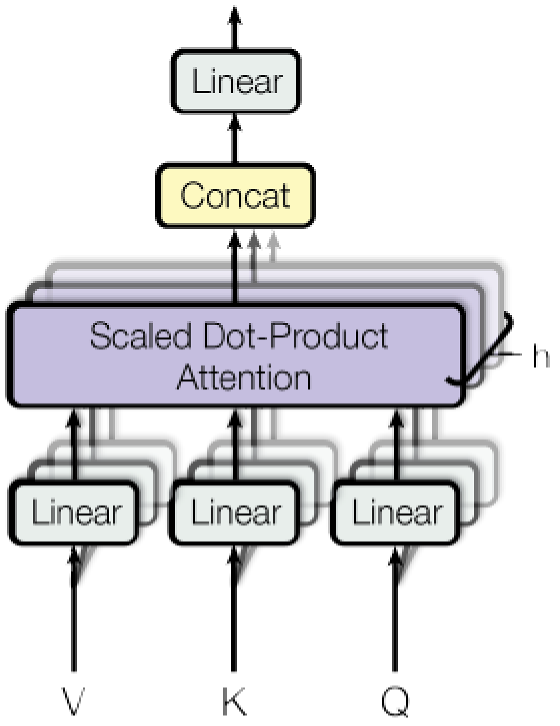
RECALL

Scaled Dot-Product Attention



$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Multi-Head Attention



$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head_1}, \ldots, \mathrm{head_h})W^O$$

$$\text{where } \mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Attend and Diagnose: Clinical Time Series Analysis Using Attention Models

Ref : http://jalammar.github.io/illustrated-transformer/

RECALL

Position-wise Feed-Forward Networks

$$\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

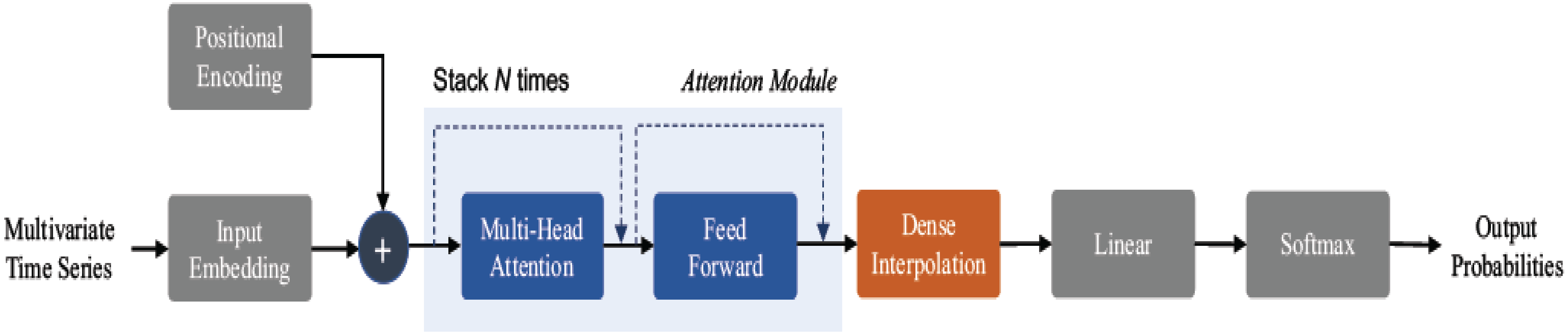
Position Encoding

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

- Make use of the order of the sequence
- Inject some information about the relative or absolute position

Attend and Diagnose: Clinical Time Series Analysis Using Attention Models
Ref : http://jalammar.github.io/illustrated-transformer/

METHOD

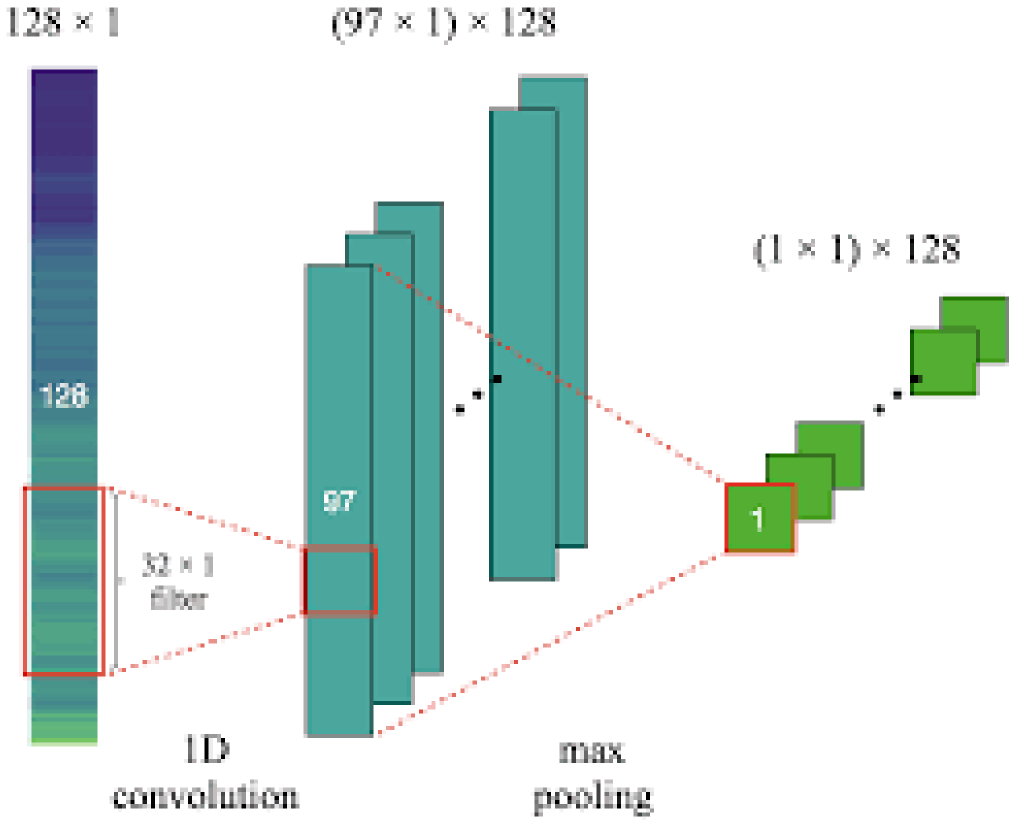


Attend and Diagnose: Clinical Time Series Analysis Using Attention Models
Ref : http://jalammar.github.io/illustrated-transformer/

METHOD

Input Embedding

Conv1d captures the dependencies across different variables without considering the temporal information



Positional Embedding

adopt sin position embedding

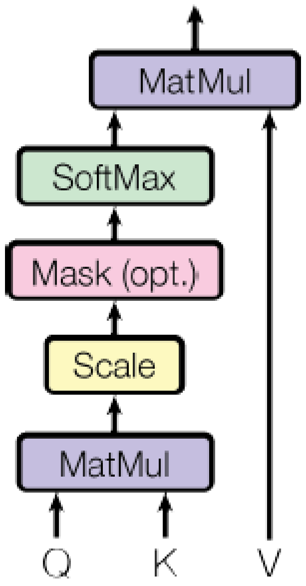
Attend and Diagnose: Clinical Time Series Analysis Using Attention Models
Ref : http://jalammar.github.io/illustrated-transformer/

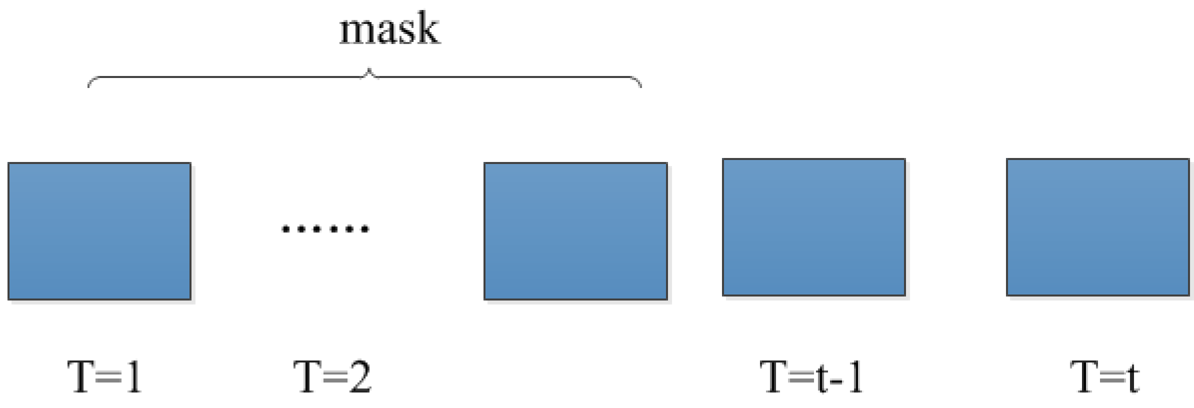
METHOD

Attention Module

Mask self-Attention

Scaled Dot-Product Attention





$$\mathrm{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\mathrm{T}}}{\sqrt{d}}\right)\mathbf{V},$$

Attend and Diagnose: Clinical Time Series Analysis Using Attention Models
Ref : http://jalammar.github.io/illustrated-transformer/

METHOD

Dense Interpolation for Encoding Order

Dense Interpolation Embedding

Input : Steps t of the time series and length of the sequence T, embeddings at step t as $\mathbf{s}_t$, factor M.

Output: Dense interpolated vector representation $\mathbf{u}$.

for $t = 1$ *to* T **do**

  $s = M * t/T$

  for $m = 1$ *to* M **do**

    $w = pow(1 - abs(s - m)/M, 2)$

    $\mathbf{u}_m = \mathbf{u}_m + w * \mathbf{s}_t$

  end

end

Algorithm 1: Dense interpolation embedding with partial order for a given sequence.

Purpose : obtain sequence representation while preserving order

w denotes the contribution of s_t to the position m of the final vector representation u

Attend and Diagnose: Clinical Time Series Analysis Using Attention Models
Modeling Order in Neural Word Embeddings at Scale

METHOD

Linear and Softmax layers

Binary classification :

$$-(y \cdot \log(\hat{y})) + (1-y) \cdot \log(1-\hat{y})$$

Multi-label classification :

$$-(y \cdot \log(\hat{y})) + (1-y) \cdot \log(1-\hat{y})$$

Regression:

$$\sum_{t=1}^{T}(l_t - \hat{l}_t)^2$$

Attend and Diagnose: Clinical Time Series Analysis Using Attention Models
Ref : http://jalammar.github.io/illustrated-transformer/

METHOD

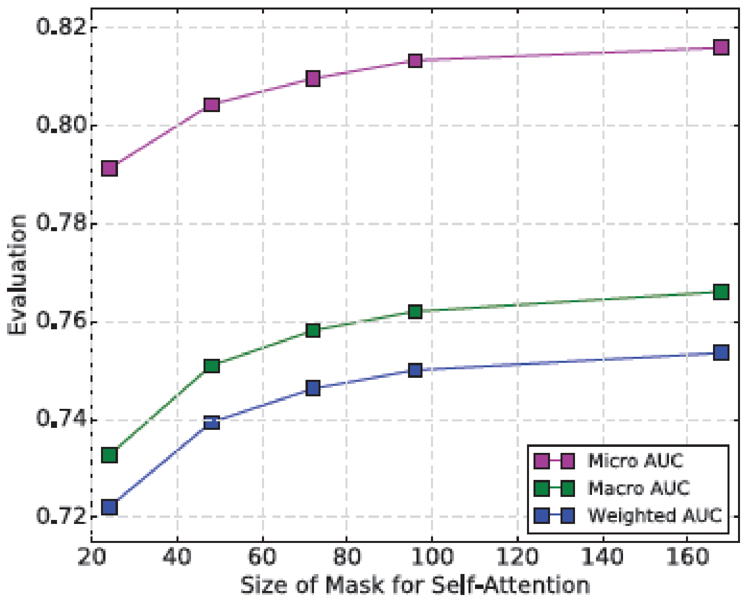
Complexity Analysis

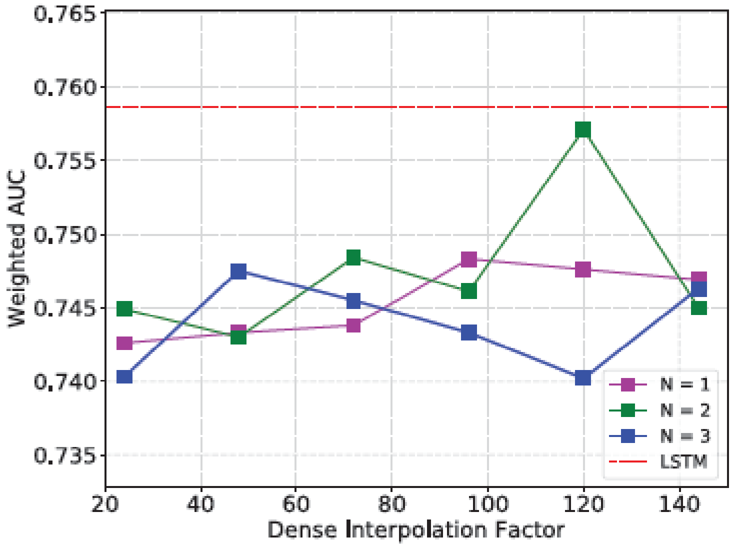
	Sequential operations	Complexity per Layer
RNN	$O(T)$	$O(T \cdot d^2)$
SAnD	$O(1)$	$O(T \cdot r \cdot d)$

Attend and Diagnose: Clinical Time Series Analysis Using Attention Models
Ref : http://jalammar.github.io/illustrated-transformer/

RESULT

Phenotyping



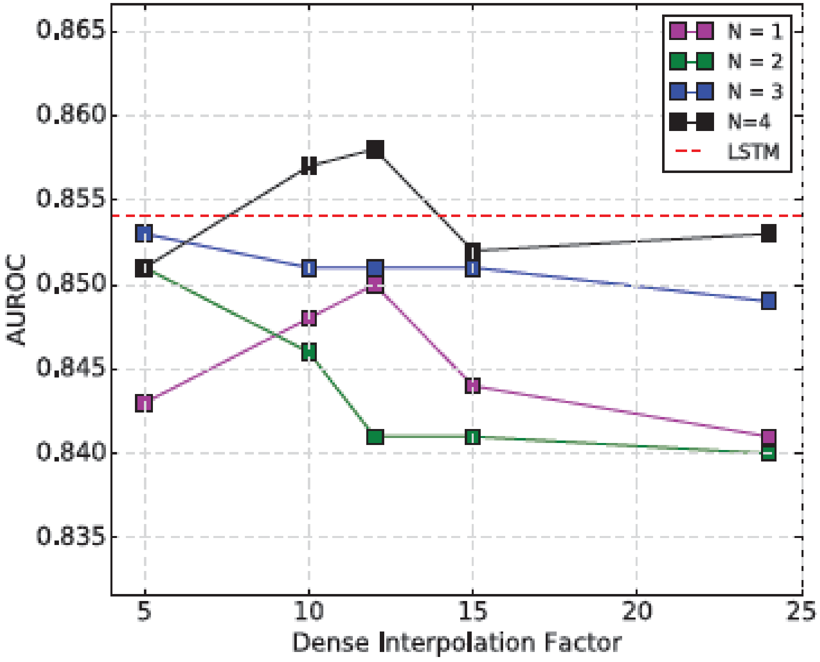


Attend and Diagnose: Clinical Time Series Analysis Using Attention Models

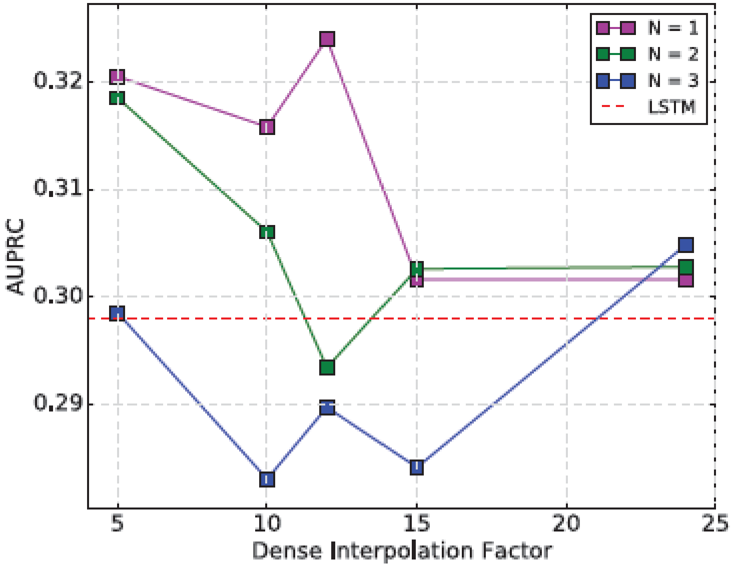
Ref : http://jalammar.github.io/illustrated-transformer/

RESULT

Mortality



Decompensation

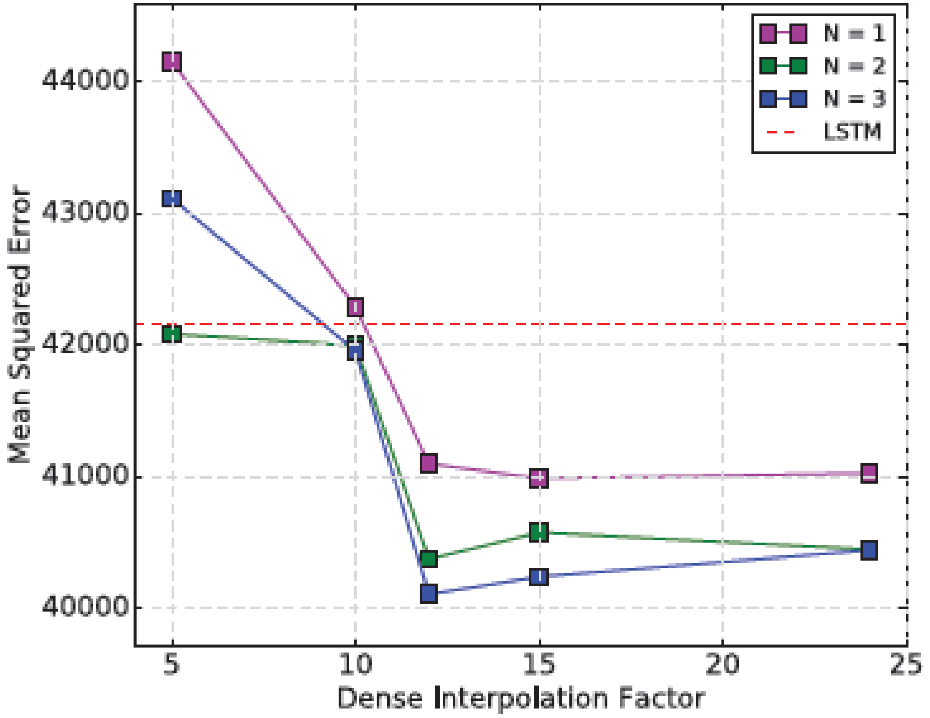


Attend and Diagnose: Clinical Time Series Analysis Using Attention Models

Ref : http://jalammar.github.io/illustrated-transformer/

RESULT

Length of Stay



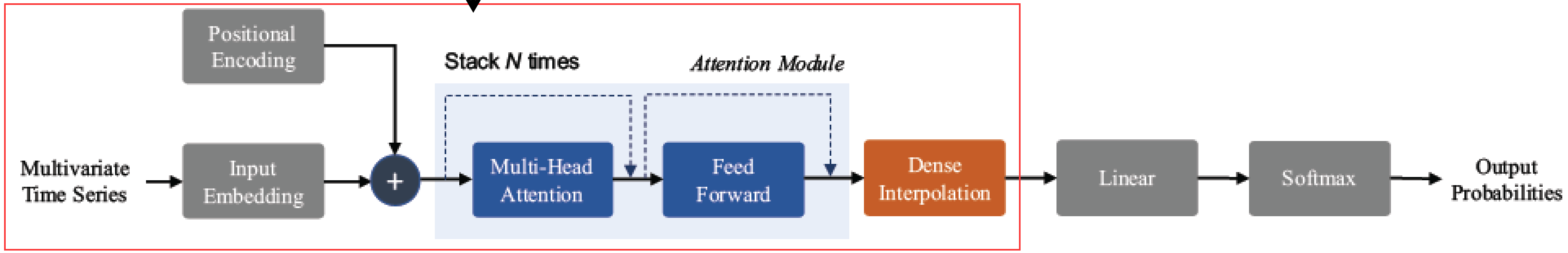
Attend and Diagnose: Clinical Time Series Analysis Using Attention Models

Ref : http://jalammar.github.io/illustrated-transformer/

RESULT

Multitask

Share Parameter



$$\ell_{mt} = \lambda_p \ell_{ph} + \lambda_i \ell_{ihm} + \lambda_d \ell_{dc} + \lambda_l \ell_{los},$$

Attend and Diagnose: Clinical Time Series Analysis Using Attention Models
Ref : http://jalammar.github.io/illustrated-transformer/

RESULT

Metrics	Method				
	LR	LSTM	*SAnD*	LSTM-Multi	*SAnD*-Multi
Task 1: Phenotyping					
Micro AUC	0.801	**0.821**	0.816	0.817	0.819
Macro AUC	0.741	**0.77**	0.766	0.766	**0.771**
Weighted AUC	0.732	0.757	0.754	0.753	**0.759**
Task 2: In Hospital Mortality					
AUROC	0.845	0.854	0.857	**0.863**	0.859
AUPRC	0.472	0.516	0.518	0.517	**0.519**
min(Se, P+)	0.469	0.491	0.5	0.499	**0.504**
Task 3: Decompensation					
AUROC	0.87	0.895	0.895	0.900	**0.908**
AUPRC	0.2132	0.298	0.316	0.319	**0.327**
min(Se, P+)	0.269	0.344	0.354	0.348	**0.358**
Task 4: Length of Stay					
Kappa	0.402	0.427	**0.429**	0.426	**0.429**
MSE	63385	42165	40373	42131	**39918**
MAPE	573.5	235.9	167.3	188.5	**157.8**

Attend and Diagnose: Clinical Time Series Analysis Using Attention Models
Ref : http://jalammar.github.io/illustrated-transformer/

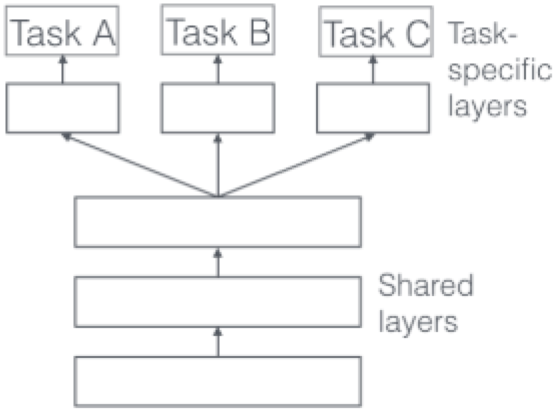
Appendix

- Hard parameter sharing

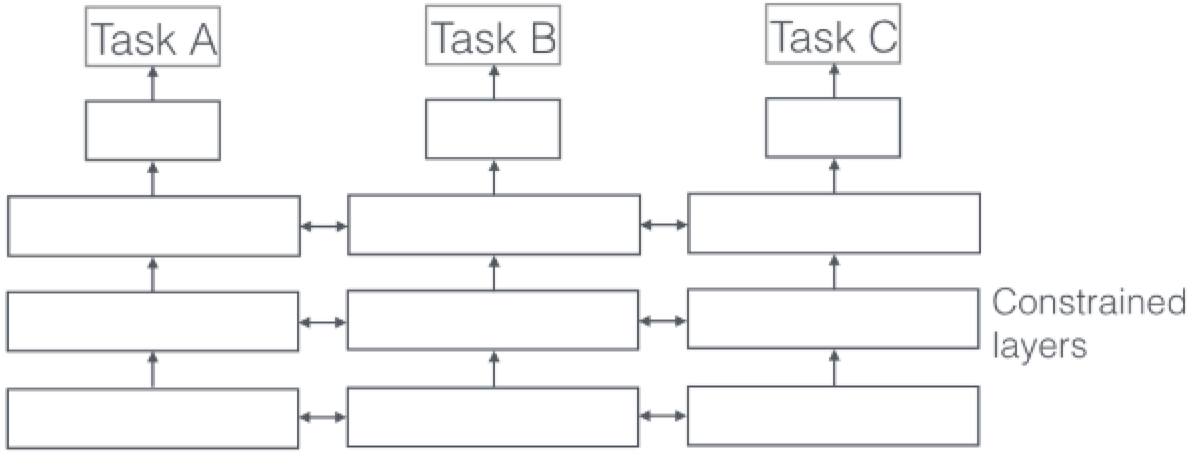
  share the hidden layers between all tasks, while keeping several task-specific output layers

- Soft parameter sharing

  each task has its own model with its own parameters, but train jointly



Hard Parameter Sharing



Soft Parameter Sharing

THANK YOU