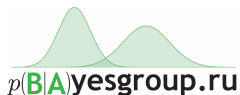


Adaptive Prediction Time for Sequence Classification Problem

Maksim Riabinin, 152
Supervisor: Ekaterina Lobacheva

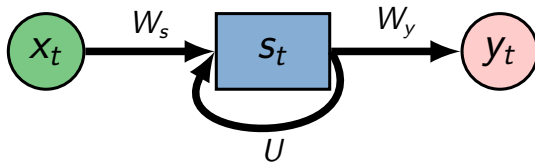
September 11, 2018



Sequence tasks in machine learning

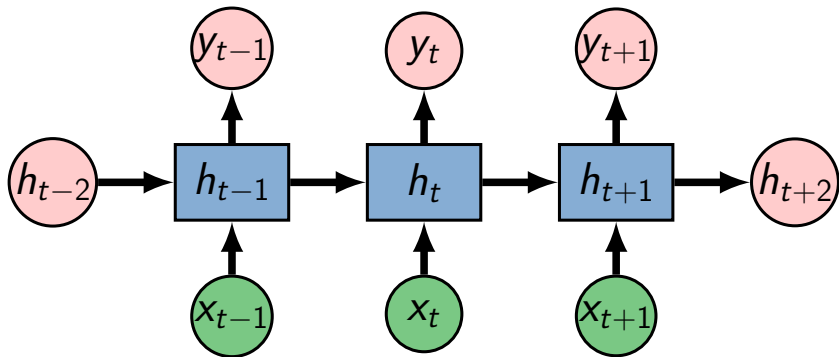
- Regular models work with fixed-size input
- If we have samples of varying size (e.g. images, sound, text), we can try to extract a fixed-size feature vector
- Such tasks are quite common:
 - Machine translation
 - Self-driving cars
 - Speech recognition
 - Clickbait text generation...
- Images can be resized, audios can be stretched...
- What can we do with text, timeseries, real world observations?
- Bag-of-words representations etc. do not account for word order
- Need to treat each step as a separate entity

RNN recap



$$s_t = \sigma_h(W_s x_t + U s_{t-1} + b_h),$$
$$y_t = \sigma_y(W_y s_t + b_y)$$

Recurrent Neural Networks (RNNs)



$$s_t = \sigma_h(W_s x_t + U s_{t-1} + b_h),$$
$$y_t = \sigma_y(W_y s_t + b_y)$$

Early prediction problem

- Regular sequence classification: $\hat{y} = a(x)$, $y \in \{1, \dots, C\}$, $x = (x_1, \dots, x_T)$ (data is separated into timesteps)
- Early classification — need to output a label as soon as possible leveraging incomplete information
- Applications: medical monitoring, self-driving cars, video surveillance
- Need to balance between contradictory objectives
- Existing methods (Santos et al., 2016) are intractable for high-dimensional data (e.g. kNN on all prefixes of a video clip)

Models with variable computation

Several approaches aim to use fewer tokens overall, which is similar to the problem in question

- REINFORCE-based models (Yu et al., 2017): probabilistic policy $p_\theta(s, a)$ results in trajectories z_i

$$J(\theta) = \mathbb{E}_{p_\theta} R_s^a,$$
$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^N \sum_{s,a \in z_i} \nabla_\theta \log p_\theta(s, a) R_s^a$$

- Straight-through estimation (Bengio et al., 2013): Skip RNN (Campos et al., 2018) as an example

$$u_t = [\hat{u}_t > 0.5], \quad \frac{\partial u_t}{\partial \hat{u}_t} = 1$$

Proposed model

T — sequence length, θ — recurrent layer parameters,

$$s_t = \text{RNN}(x_t, s_{t-1}, \theta), \quad h_t = \sigma(W_h s_t + b_h),$$

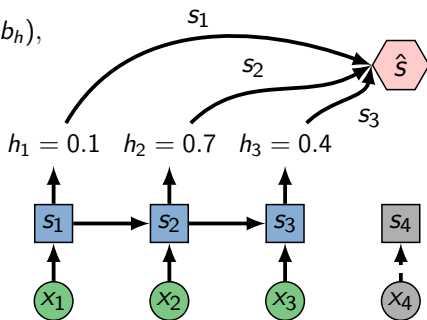
$$p_t = \begin{cases} R, & t = N, \\ h_t, & t < N, \end{cases}$$

$$N = \min \left\{ t' : \sum_{t=1}^{t'} h_t \geq 1 - \varepsilon, \quad T \right\},$$

$$R = 1 - \sum_{t=1}^{N-1} h_t,$$

$$\hat{s} = \sum_{t=1}^N p_t s_t$$

We use GRU (Cho et al., 2014) in our experiments, although variations are possible



$$\begin{aligned} p_1 &= 0.1, & p_2 &= 0.7, \\ p_3 &= R = 0.2, \\ p_4 &= 0, & N &= 3 \end{aligned}$$

Encouraging early prediction

- N is not differentiable w.r.t. model parameters
- Can't optimize directly
- We add remainder as a penalty term instead

$$\hat{L}(y, \hat{y}) = L(y, \hat{y}) + \lambda R$$

$$\frac{\partial \hat{L}}{\partial h_t} = \frac{\partial L}{\partial h_t} + \lambda \cdot \begin{cases} 0, & t = N, \\ -1, & t < N \end{cases}$$

This increases halting scores of all steps except the last, making the computation stop earlier

Experiment setup and baselines

1. Regular GRU without early stopping — measure quality loss from using fewer tokens
2. Skip RNN (Campos et al., 2018) — compare overall computational cost reduction
3. REINFORCE (Williams, 1992) with discrete stopping decision (sample Bernoulli random variable for every timestep), reward is negative number of steps, entropy regularization and learned baselines as additional improvements

We use RMSprop optimizer with learning rate 10^{-3} and $\epsilon = 0.01$. Gradient norm is clipped to 1, initial RNN state is also trained. Training is performed for 600 epochs with batch size of 512

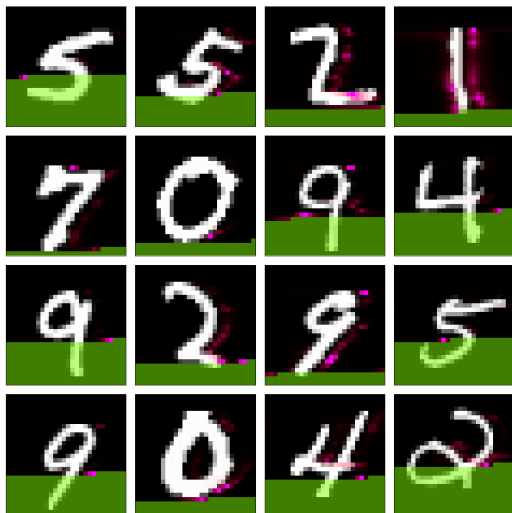
Sequential MNIST

Handwritten 28x28 digits represented as black and white images
To test model capability to work with long sequences, we feed images pixel by pixel into the network

Model	Accuracy	Steps	Prefix length
GRU	97.4	784	784
Skip GRU	97.5	393.78	783.18
REINFORCE, $\alpha = 0.05$, $\beta = 0.05$	88.94	313.4	313.4
REINFORCE, $\alpha = 0$, $\beta = 0$	98.69	784	784
APT, $\lambda = 10^{-2}$	98.52	536.2	536.2
APT, $\lambda = 1.5 \cdot 10^{-3}$	98.65	635.1	635.1
APT, $\lambda = 0$	98.83	658.8	658.8

Halting probabilities

Magnitude of p_t indicated by red color, pixels in green are not used



Samples are treated differently based on their contours, similar to attention (Bahdanau et al., 2015)

ReorderedMNIST

Same digits, but pixels are arranged in descending order of variance over the training dataset

Model	Accuracy	Steps	Prefix length
GRU	11.32	784	784
Skip GRU	89.45	282.28	774.12
REINFORCE, $\alpha = 0.05$, $\beta = 0.05$	86.84	275.5	275.5
REINFORCE, $\alpha = 0.01$, $\beta = 0.01$	87.43	714.8	714.8
APT, $\lambda = 10^{-2}$	90.15	262.4	262.4
APT, $\lambda = 1.5 \cdot 10^{-3}$	90.34	272.2	272.2
APT, $\lambda = 0$	90.86	324.4	324.4



UCF-101

Videos of 101 different actions at resolution 320x240

Each video is treated as a sequence of up to 250 frames, activations from ResNet-50 pretrained on ImageNet are used as 2048-dimensional feature representations

Model	Accuracy	Steps	Prefix length
GRU	81.7	250	250
Skip GRU	79.2	29.7	—
APT, $\lambda = 10^{-5}$	79.3	85.43	85.43



Billiards



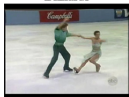
Cliff-diving



Cricket Shot



Field Hockey Penalty



Ice dancing



Javelin throw



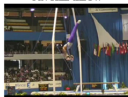
Pizza tossing



Playing Cello



Soccer Juggling



Still Rings



Sumo Wrestling







Writing-on-board

Conclusion and future work

- A technique inspired by Adaptive Computation Time (Graves, 2016), but used for tangentially related task
- Conducted experiments on several datasets show that model maintains classification accuracy and uses short prefix of input
- Possible to combine with other steps number reduction methods

References I

-  Bahdanau, Dzmitry et al. (2015). “Neural machine translation by jointly learning to align and translate”. In: *International Conference on Learning Representations*.
-  Bengio, Yoshua et al. (2013). *Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation*. eprint: [arXiv:1308.3432](https://arxiv.org/abs/1308.3432).
-  Campos, Víctor et al. (2018). “Skip RNN: Learning to Skip State Updates in Recurrent Neural Networks”. In: *International Conference on Learning Representations*.
-  Cho, Kyunghyun et al. (2014). “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation”. In: *Conference on Empirical Methods in Natural Language Processing*.
-  Graves, Alex (2016). “Adaptive Computation Time for Recurrent Neural Networks”. In: *CoRR* abs/1603.08983. [arXiv: 1603.08983](https://arxiv.org/abs/1603.08983).

References II



Santos, Tiago et al. (2016). “A Literature Survey of Early Time Series Classification and Deep Learning”. In: *SAMI@iKNOW*.



Williams, Ronald J. (May 1992). “Simple statistical gradient-following algorithms for connectionist reinforcement learning”. In: *Machine Learning* 8.3, pp. 229–256. ISSN: 1573-0565. DOI: 10.1007/BF00992696.



Yu, Adams Wei et al. (2017). “Learning to Skim Text”. In: *Annual Meeting of the Association for Computational Linguistics*.