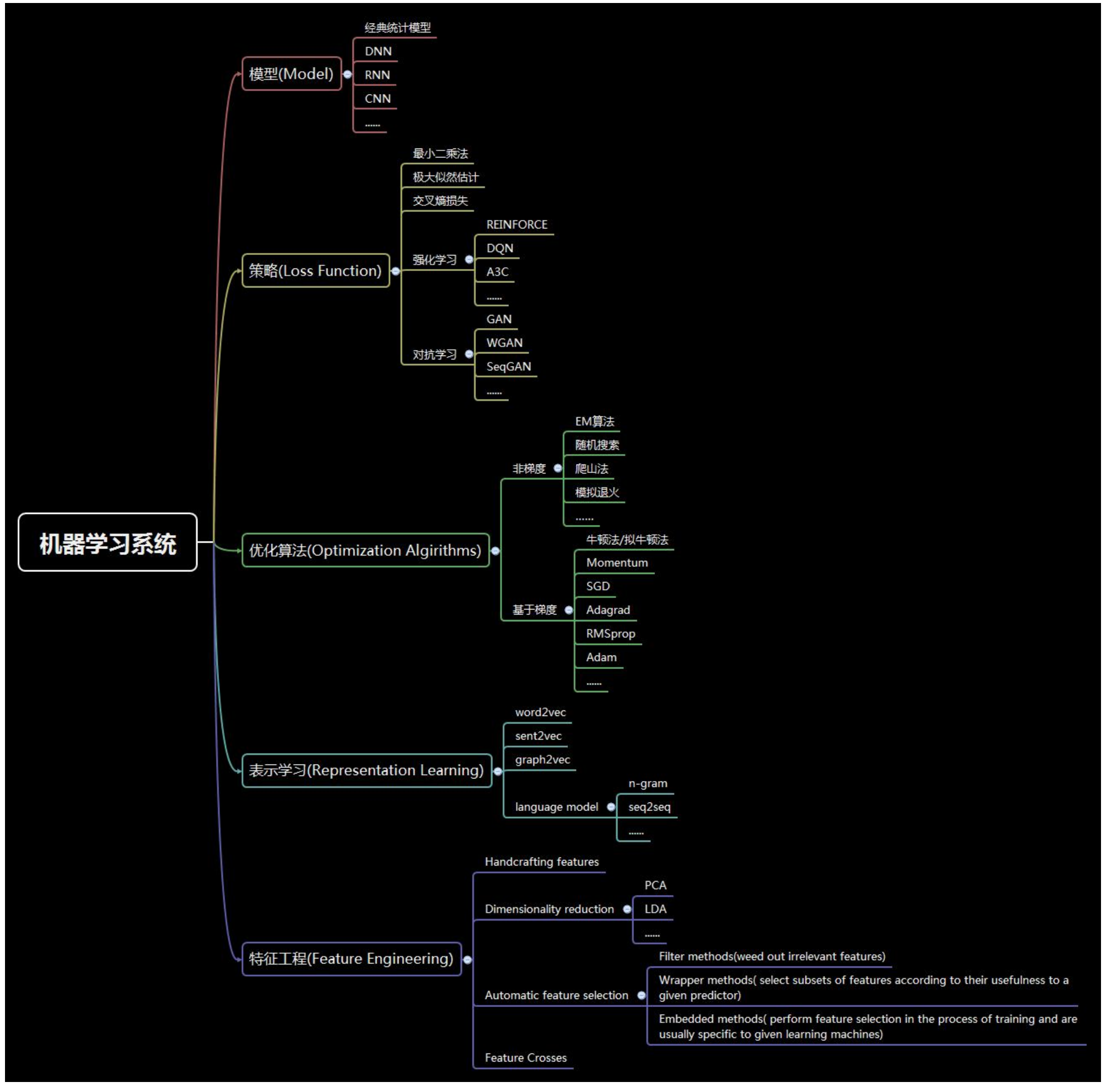
机器学习与RMT结合探索

姜衡军(北京邮电大学计算机系)

大纲

- **机器学习系统概览**
- 我的NLP研究工作回顾
    - 端到端对话系统
    - 序列标注
    - 动态激活函数选择
    - 数据增强
- 机器学习与RMT结合探索

机器学习系统概览



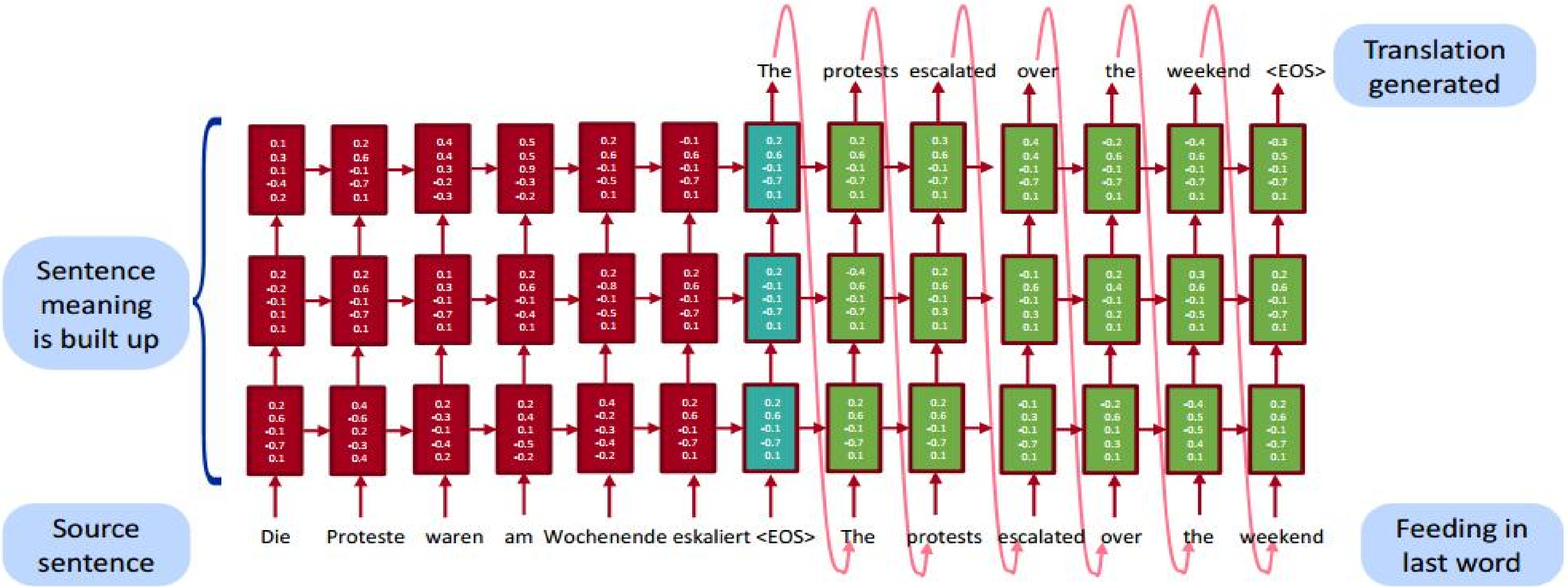
机器学习系统.PNG

大纲

- 机器学习系统概览
- **我的NLP研究工作回顾**
  - **端到端对话系统**
  - **序列标注**
  - **动态激活函数选择**
  - **数据增强**
- 机器学习与RMT结合探索

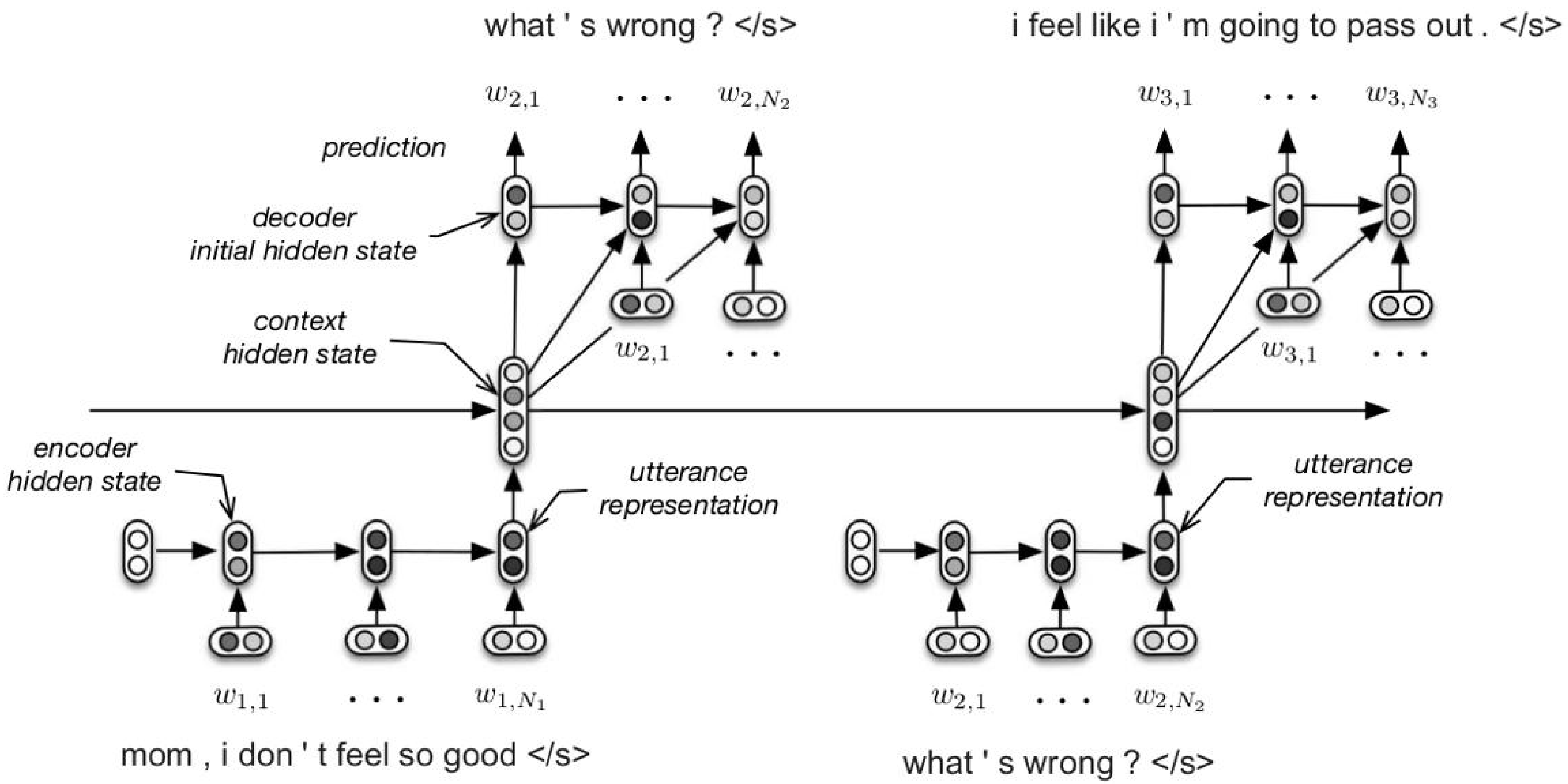
端到端对话系统

序列到序列模型：



方法与模型

层次化编解码模型：



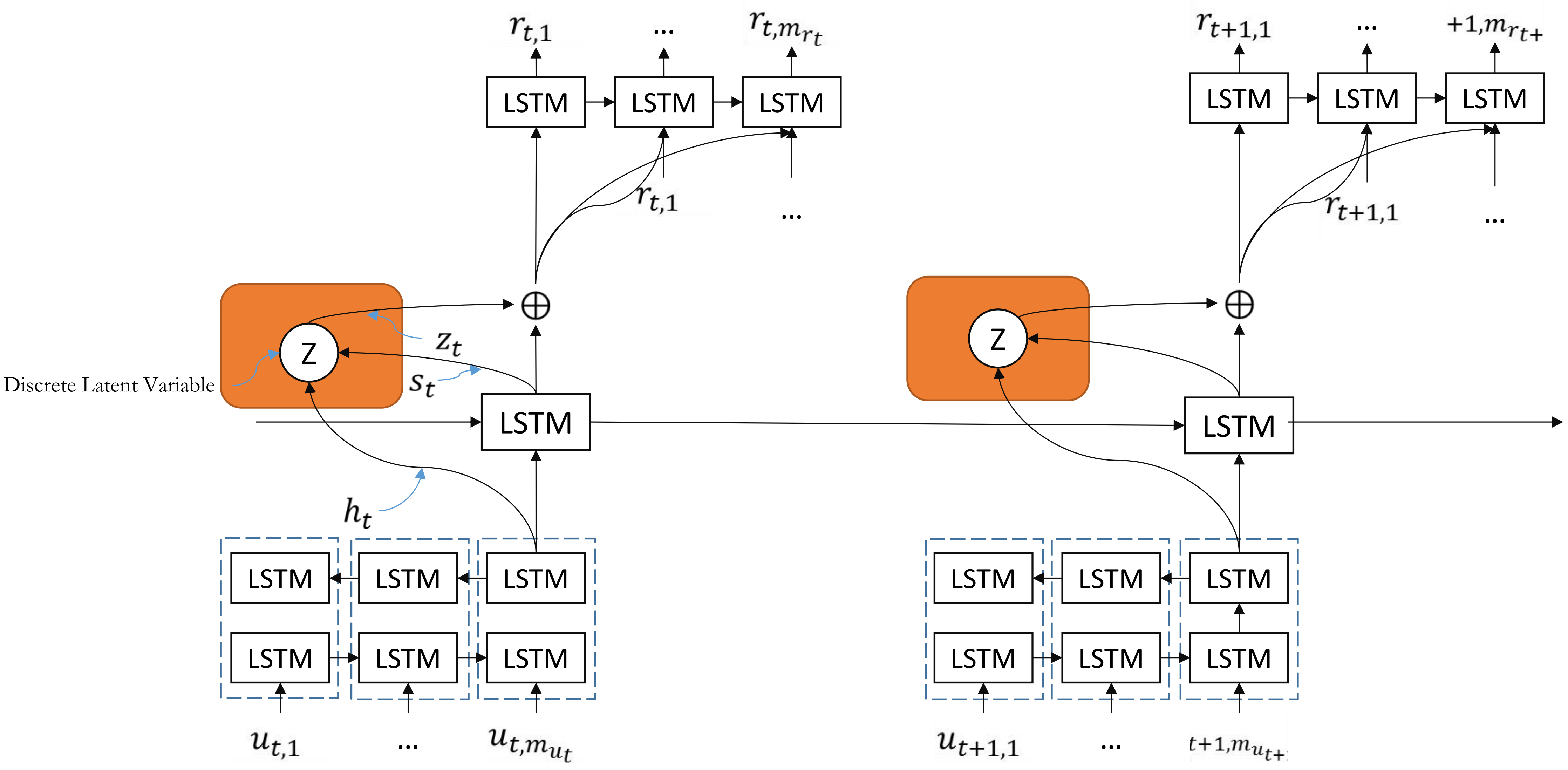
方法与模型

引入基于**离散隐变量**的层次化编解码模型（DVHRED）：

基本原理：

在层次化编解码结构基础上引入离散隐变量，从而更好地对文本语义及用户意图进行建模

方法与模型



$r_{t,1}$
...
$r_{t,m_{r_t}}$
$r_{t+1,1}$
...
$+1,m_{r_{t+}}$
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
$r_{t,1}$
...
$r_{t+1,1}$
...
Z
z_t
s_t
Z
Discrete Latent Variable
LSTM
LSTM
h_t
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
$u_{t,1}$
...
$u_{t,m_{u_t}}$
$u_{t+1,1}$
...
$t+1,m_{u_{t+}}$

变分贝叶斯推断

潜在用户意图变量: z
近似后验概率分布: q(z|x)
真实后验概率分布: p(z|x)

$$\min KL(q \| p) = \int q(z|x) \log \frac{q(z|x)}{p(z|x)} dz$$

$$= -\int q(z|x) \log \frac{p(z|x)}{q(z|x)} dz$$

$$= -\int q(z|x) \log \frac{p(z,x)}{q(z|x)p(x)} dz$$

$$= \int q(z|x)[\log q(z|x) + \log p(x)]dz - \int q(z|x) \log p(z,x) dz$$

$$= \log p(x) + \int q(z|x) \log q(z|x) dz - \int q(z|x) \log p(z,x) dx$$

$$=> \log p(x) = KL(q \| p) + L(q)$$

$$=> \min KL(q \| p) == \max L(q)[\# ELOB(\text{Evidence Lower Bound})]$$

$$L(q) = \int q(z|x) \log p(z,x) dz - \int q(z|x) \log q(z|x) dz$$

$$= \int \mathrm{q}(z|x) \log p(x|z) dz + \int q(z|x) \log p(z) dz - \int q(z|x) \log q(z|x) dz$$

$$= \int q(z|x) \log p(x|z) dz - \int q(z|x) \log \frac{q(z|x)}{p(z)} dz$$

$$= \int q(z|x) \log p(x|z) dz - KL[q(z|x) \| p(z)]$$

方法与模型

$$\mathrm{h}_t = biLSTM(u_t)$$

$$\mathrm{s}_t = LSTM(s_{t-1}, h_t)$$

$$\mathrm{z}_{\mathrm{t}}^{1} \sim \pi_\theta(z_t \mid s_t, h_t)$$

方法与模型

由于上述模型缺少**领域知识**，因此只能与用户进行“闲聊”

User: I want to order a chinese restaurant.

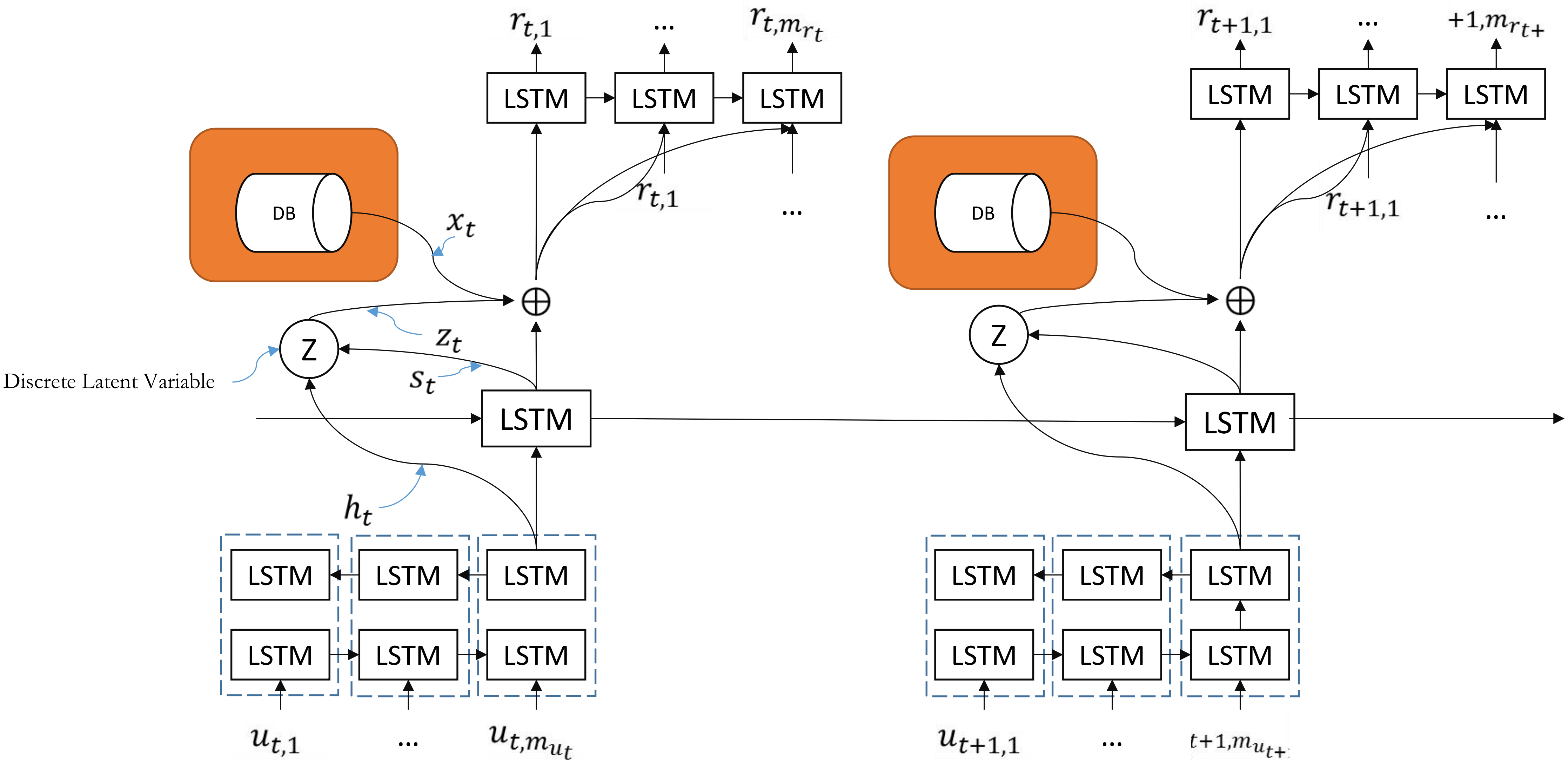
Bot: sounds a good idea.

为了更好的完成任务型对话，需要在DVHRED框架上加入知识库（Knowledge Base, KB）(DVHRED+KB)

User: I want to order a chinese restaurant.

Bot: The good luck chinese food takeaway is in the south area.

方法与模型



$r_{t,1}$
...
$r_{t,m_{r_t}}$
$r_{t+1,1}$
...
$+1,m_{r_{t+}}$
LSTM
DB
x_t
$r_{t,1}$
$r_{t+1,1}$
z_t
s_t
Z
Discrete Latent Variable
h_t
$u_{t,1}$
$u_{t,m_{u_t}}$
$u_{t+1,1}$
$t+1,m_{u_{t+}}$

方法与模型

x_t : 10维的向量，[0, 0, 0, 0, 0, 1, 0, 1, 0, 1]

分别表示['phone_request'，'postcode_request', 'address_request', 'name_other', 'food_request', 'food_inform', 'pricerange_request', 'pricerange_inform', 'area_request', 'area_inform']

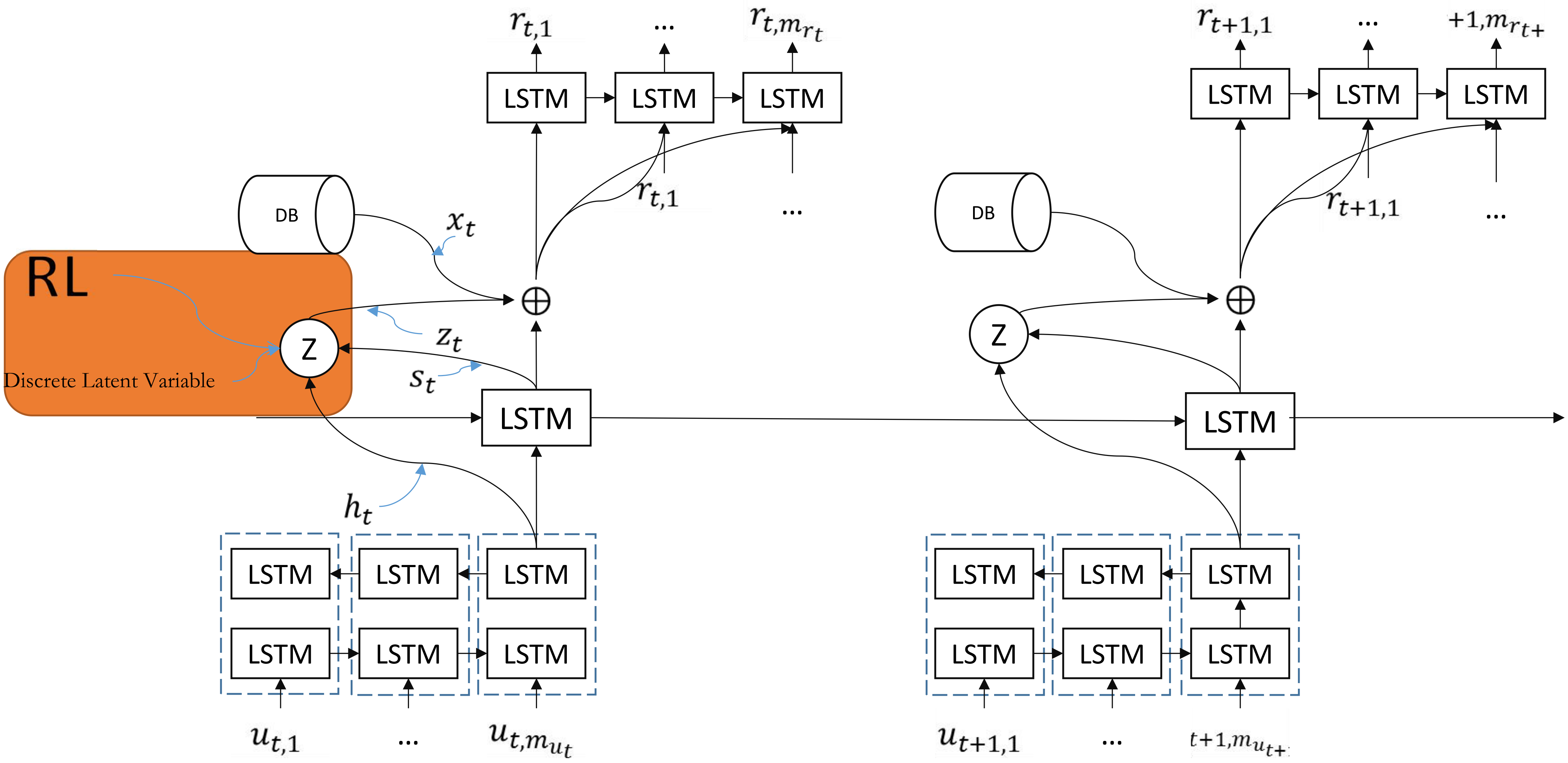
方法与模型

进一步，为了提升任务型对话的成功率，需要在选用户潜在意图时加入明确的激励

因此，我们在DVHRED+KB的基础上，通过强化学习，引入奖励函数

$$r_t = \eta \cdot \mathrm{sBLEU}(m_t, \hat{m}_t) + \begin{cases} 1 & m_t \text{ improves} \\ -1 & m_t \text{ degrades} \\ 0 & \text{otherwise} \end{cases}$$

方法与模型



$r_{t,1}$
...
$r_{t,m_{r_t}}$
$r_{t+1,1}$
...
$+1,m_{r_{t+}}$
LSTM
DB
x_t
$r_{t,1}$
...
$r_{t+1,1}$
RL
Z
z_t
s_t
Discrete Latent Variable
h_t
$u_{t,1}$
...
$u_{t,m_{u_t}}$
$u_{t+1,1}$
$t+1,m_{u_{t+}}$

REINFORCE Algorithm

$$\max E_{\pi}[\sum_{k}\gamma^{k} r_{t+k} \mid s_t = s] = \max \sum_{a_t} P_{\theta}(a_t \mid s_t;\theta) * Q(s_t, a_t)$$

$$l(\theta) = -\sum_{a_t} P_{\theta}(a_t \mid s_t;\theta) * Q(s_t, a_t)$$

$$\nabla_{\theta} l(\theta) = -\sum_{a_t} \nabla_{\theta} P_{\theta}(a_t \mid s_t;\theta) * Q(s_t, a_t)$$

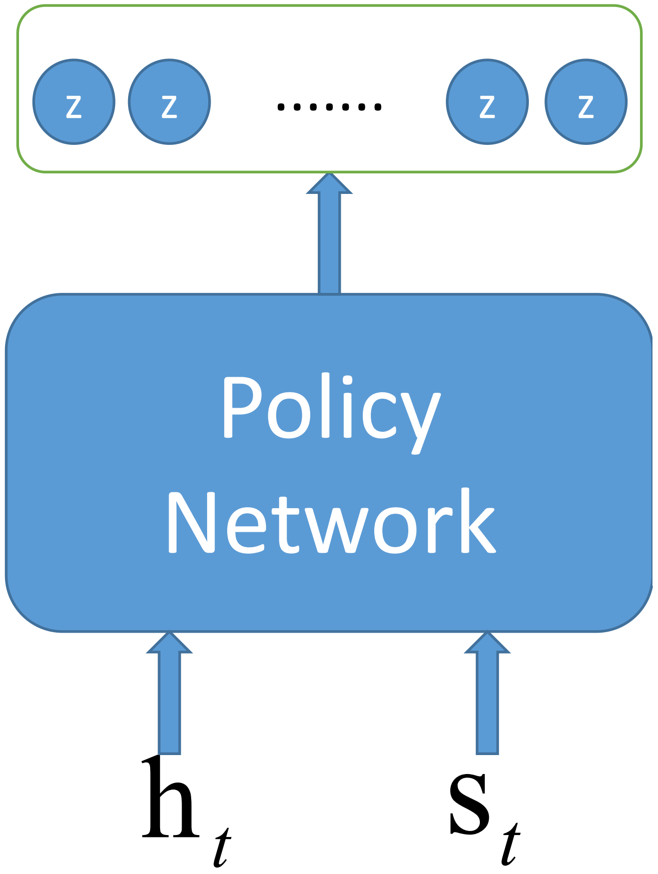
$$= -\sum_{a_t} P_{\theta}(a_t \mid s_t;\theta) * \nabla_{\theta} \log(P_{\theta}(a_t \mid s_t;\theta)) * Q(s_t, a_t)$$

$$= -E_{\pi}[\nabla_{\theta} \log(P_{\theta}(a_t \mid s_t;\theta)) * Q(s_t, a_t) \mid \mathrm{s}_t]$$

$$L\ (\theta) = -\frac{1}{N}\sum_{n=1}^{N}\ \sum_{t=1}^{T} \log(P_{\theta}(a_t^n \mid \mathrm{s}_\mathrm{t};\theta)) * Q(\mathrm{s}_\mathrm{t}, a_t^n) \qquad (1) \qquad Q(s_t, a_t) = E\ [\sum_{k=0}^{T}\gamma^{k} r_{t+k} \mid s_t, a_t] \qquad (2)$$

方法与模型

$$z_t \sim \pi_\theta(z \mid h_t, s_t)$$



Step 1.

加载 pre-trained 模型

Step 2.

微调策略网络

For t in all_turn do

For m in M do

从策略网络中采样 $z_t^m \sim \pi_{\Theta_2}(z \mid s_t, h_t)$

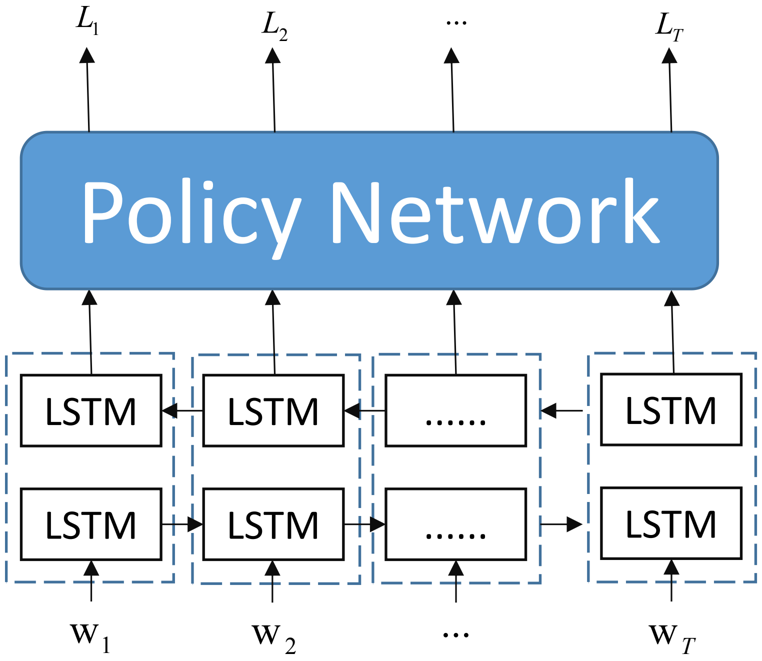
End for

更新策略网络 π_{Θ_2} 参数：

$$\frac{\partial J}{\partial \Theta_2} \approx \frac{1}{M}\sum_{m=1}^{M} R_t^m \frac{\partial \log \pi_{\Theta_2}(z_t^m \mid s_t, h_t)}{\partial \Theta_2}$$

End for

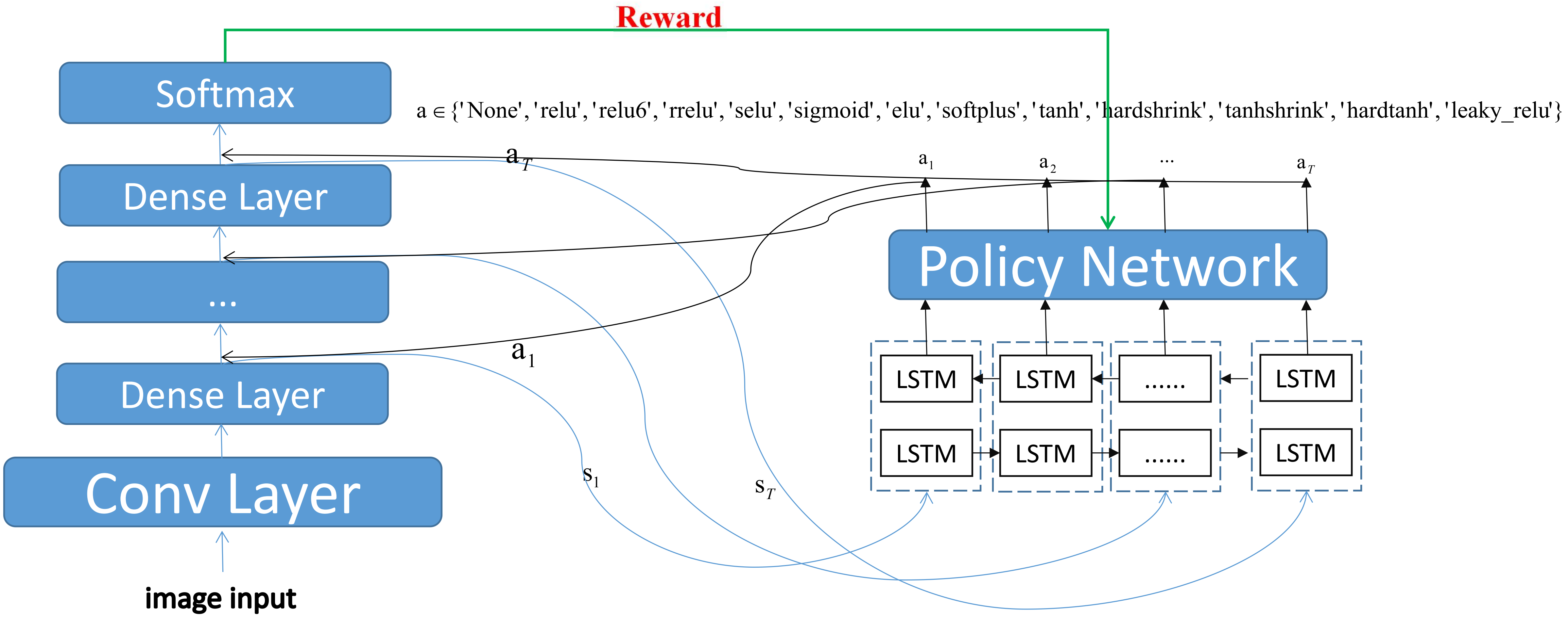
序列标注



$$L(\theta) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T}\log(P_\theta(L_t^n \mid s_t;\theta)) * Q(s_t, L_t^n) \quad (3)$$

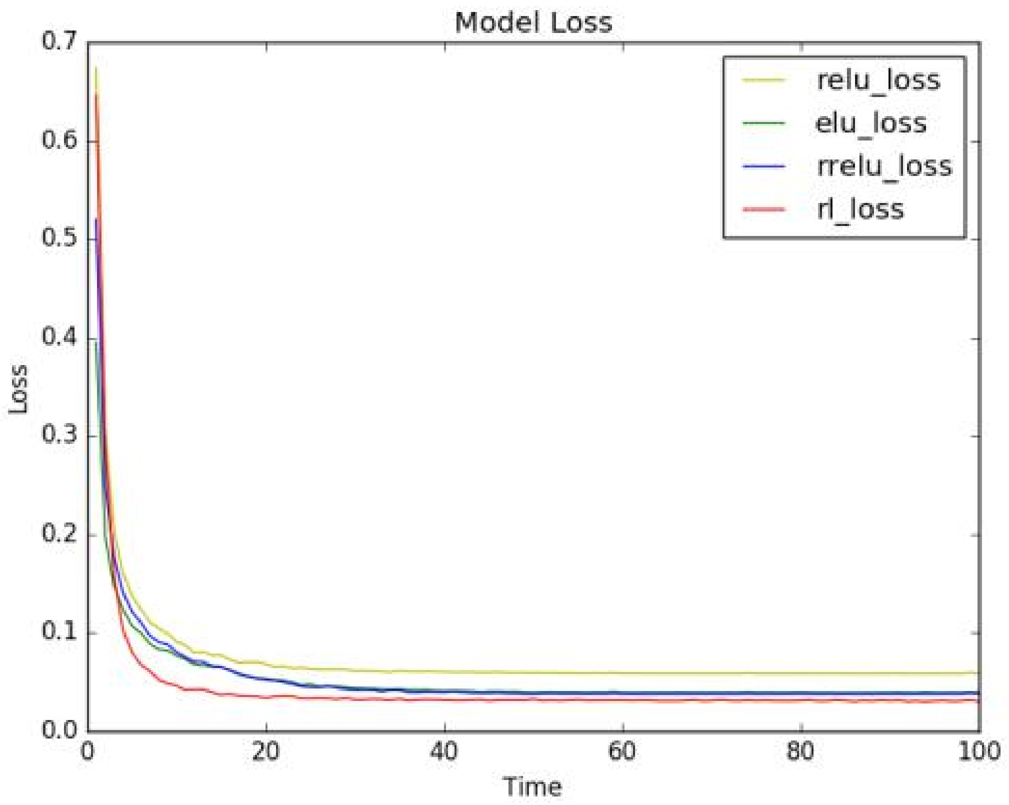
$$Q(s_t, L_t) = E\,[\sum_{k=0}^{T}\gamma^k r_{t+k} \mid s_t, L_t] \quad (4)$$

动态激活函数选择

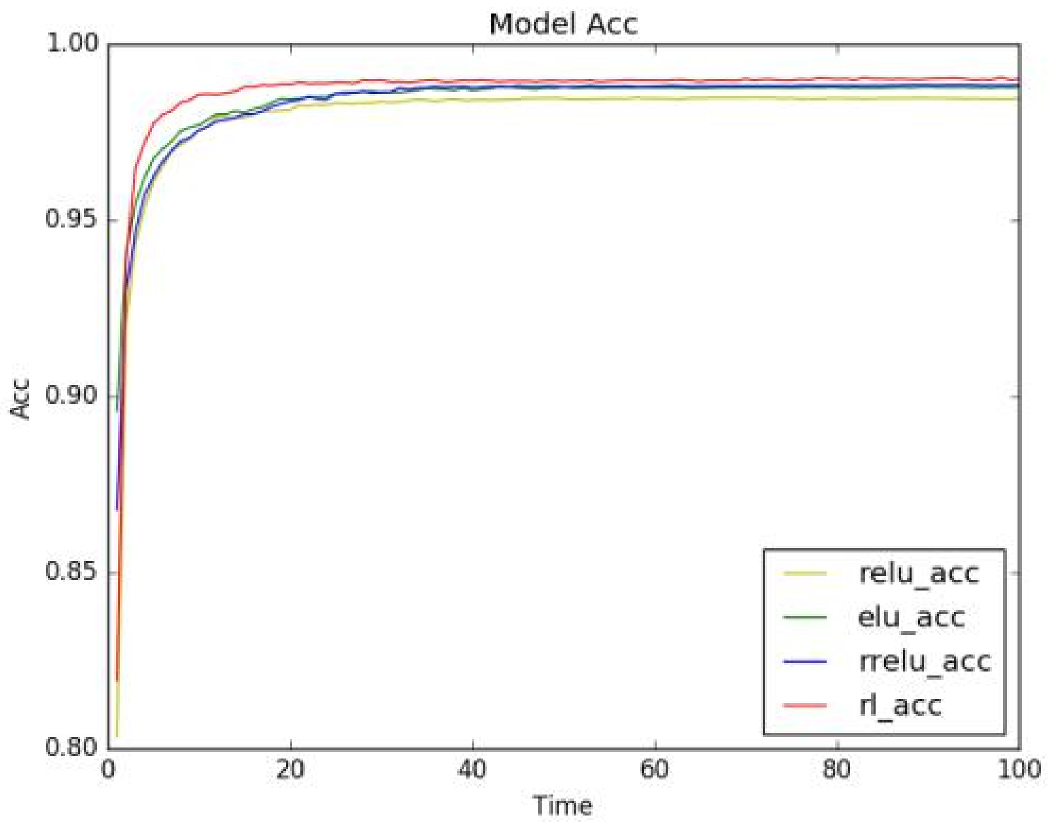


Reward
Softmax
Dense Layer
...
Dense Layer
Conv Layer
image input
a ∈ {'None','relu','relu6','rrelu','selu','sigmoid','elu','softplus','tanh','hardshrink','tanhshrink','hardtanh','leaky_relu'}
a_T
a_1
s_1
s_T
a_1
a_2
...
a_T
Policy Network
LSTM
LSTM
......
LSTM
LSTM
LSTM
......
LSTM

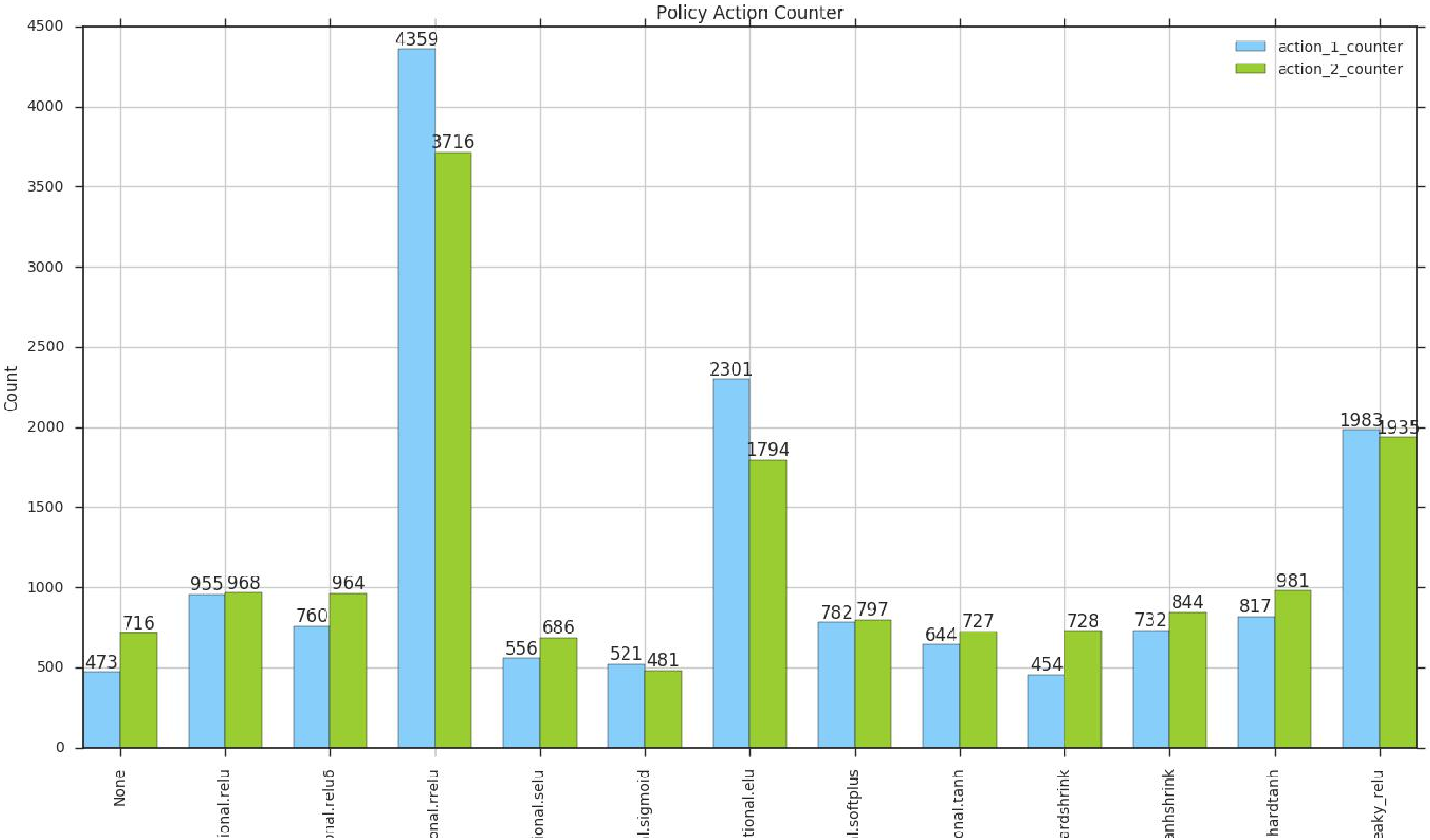
结果



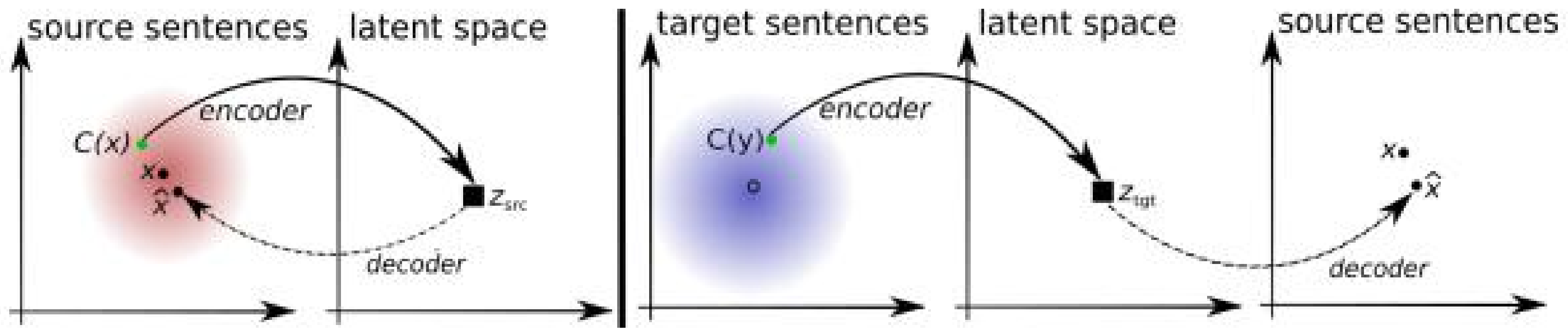
Model Loss
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Loss
0
20
40
60
80
100
Time
relu_loss
elu_loss
rrelu_loss
rl_loss

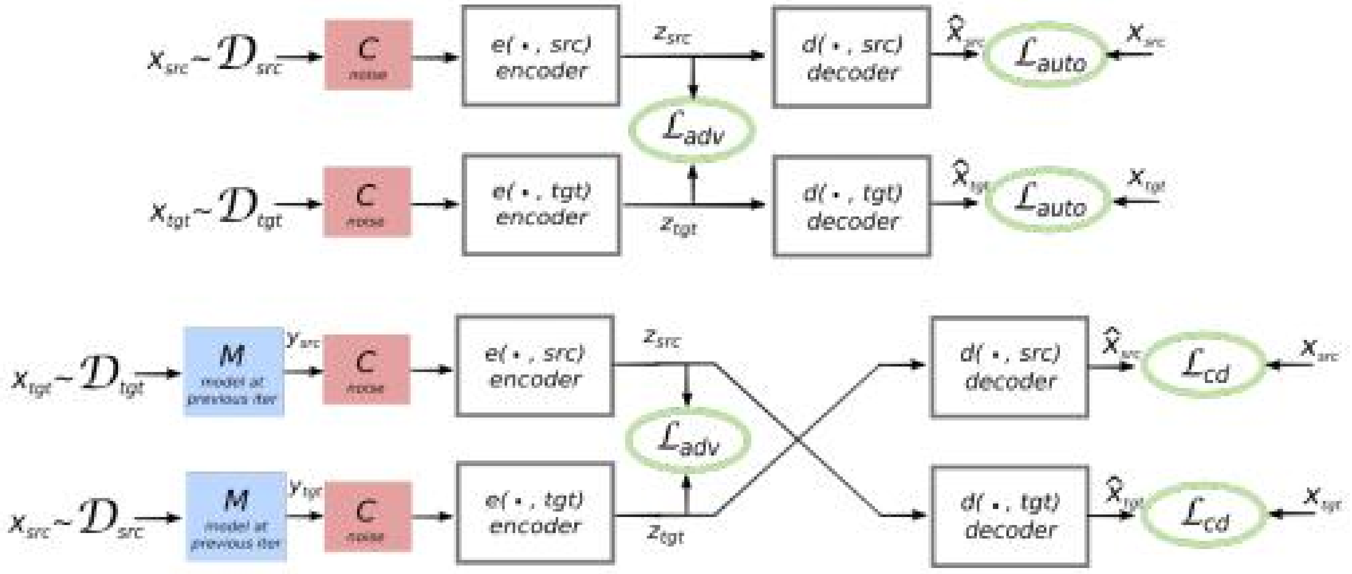


Model Acc
1.00
0.95
0.90
0.85
0.80
Acc
0
20
40
60
80
100
Time
relu_acc
elu_acc
rrelu_acc
rl_acc


Policy Action Counter
action_1_counter
action_2_counter
Count
4500
4000
3500
3000
2500
2000
1500
1000
500
0
473
716
955
968
760
964
4359
3716
556
686
521
481
2301
1794
782
797
644
727
454
728
732
844
817
981
1983
1935
None
ional.relu
onal.relu6
onal.rrelu
ional.selu
l.sigmoid
tional.elu
l.softplus
onal.tanh
ardshrink
anhshrink
hardtanh
eaky_relu

数据增强


source sentences
latent space
target sentences
latent space
source sentences
encoder
encoder
C(x)
C(y)
decoder
decoder


e(• , src) encoder
d(• , src) decoder
e(• , tgt) encoder
d(• , tgt) decoder
C noise
M model at previous iter

数据增强

$$L(\theta_{enc},\theta_{dec},Z_{emb})=\lambda_{auto}[L_{auto}(\theta_{enc},\theta_{dec},Z_{emb},src)+L_{auto}(\theta_{enc},\theta_{dec},Z_{emb},tgt)]+$$
$$\lambda_{cd}[L_{cd}(\theta_{enc},\theta_{dec},Z_{emb},src,tgt)+L_{cd}(\theta_{enc},\theta_{dec},Z_{emb},tgt,src)]+$$
$$\lambda_{adv}L_{adv}(\theta_{enc},Z_{emb}\mid\theta_{D})$$

where

$$L_{auto}(\theta_{enc},\theta_{dec},Z_{emb},l)=E_{x\sim D_l,\tilde{x}\sim d(e(C(x),l),l)}[\Delta(\tilde{x},x)]$$

$$L_{cd}(\theta_{enc},\theta_{dec},Z_{emb},l_1,l_2)=E_{x\sim D_{l_1},\tilde{x}\sim d(e(C(M(x)),l_2),l_1)}[\Delta(\tilde{x},x)]$$

$$L_{adv}(\theta_{enc},Z_{\mathrm{emb}}\mid\theta_{D})=-E_{(x_i,l_i)}[\log p_D(l_j\mid e(x_i,l_j))]$$

$$L_D(\theta_D\mid\theta,Z)=-E_{(x_i,l_i)}[\log p_D(l_i\mid e(x_i,l_i))]$$

数据增强

Algorithm 1 Unsupervised Training for Machine Translation

procedure TRAINING($\mathcal{D}_{src}, \mathcal{D}_{tgt}, T$)
  Infer bilingual dictionary using monolingual data (Conneau et al., 2017)
  $M^{(1)} \leftarrow$ unsupervised word-by-word translation model using the inferred dictionary
  for $t = 1, T$ **do**
    using $M^{(t)}$, translate each monolingual dataset
    // discriminator training & model training as in eq. 4
    $\theta_{\text{discr}} \leftarrow \arg\min \mathcal{L}_D, \quad \theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z} \leftarrow \arg\min \mathcal{L}$
    $M^{(t+1)} \leftarrow e^{(t)} \circ d^{(t)}$ // update MT model
  end for
  return $M^{(T+1)}$
end procedure

结果

- 帮我预订个<入住城市>旁边的<酒店品牌>，<入住日期>入住 END | 帮我预订个<入住城市><地址>周边的酒店，<入住日期>入住
- 我打算查下<出发时间>从<出发城市>出发的车票 END | 我打算查下从<出发城市>出发的<列车类型><座位类型>，<出发日期>的
- 搜下<日期><城市>限行状况如何 END | 搜下<城市><日期>的限行情况
- 给我设定个<日期>的起床闹钟哦 END | 给我设定一个起床闹钟，<日期>的哦
- 我想要把<全部范围>的<设备名>亮度小一点 END | 我想要把<房间><全部范围>的<设备名>亮度小一点
- <出发城市>到<到达城市>车票还有吗，<出发日期>的 END | 我有没有<出发城市>到<到达城市>票
- 帮我看一下概念是什么 END | 帮我看一下<定义关键词>的概念是什么
- 把<全部范围><房间><设备名>速度调高 END | 把<房间>的<设备名>风速调高
- 我要让<房间><设备名><全部范围>风速不够小小点 END | 我要让<房间>的<设备名>风速小点
- 我希望设定个<日期><时间区间>的早起闹钟 END | 我希望设定一个<日期><时间区间><时间>的早起闹钟
- 我要放一下这一个<故事内容名>的故事 END | 我要放一个这一段<故事内容名>的故事
- 将<成语>的释义，近义词，反义词都讲一下 END | 请帮我查一下<成语>的释义和意思相似的成语
- 请为我设置<时间>的闹表 END | 请为我设置个闹铃

大纲

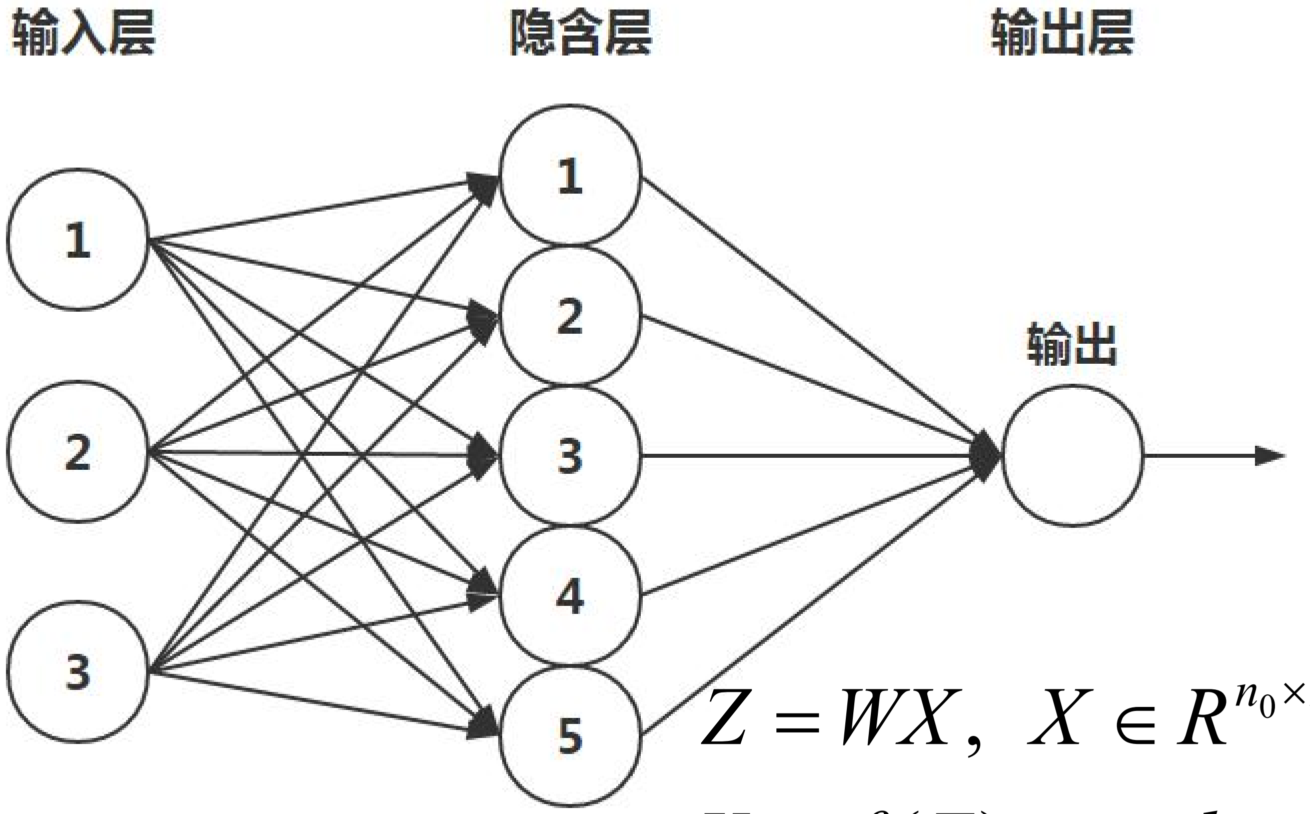
- 机器学习系统概览
- 我的NLP研究工作回顾
    - 端到端对话系统
    - 序列标注
    - 动态激活函数选择
    - 数据增强
- **机器学习与RMT结合探索**

机器学习与RMT结合探索

Borrow the idea from the statistical physics: approximate the constituents of a large complex system with random variables

- 随机特征(Random Feature)
- Hessian/Jacobian matrix of the loss function

Basics of Neural Network



$$Z = WX, \ \ X \in R^{n_0 \times m} \ W \in R^{n_1 \times n_0}$$

$$Y = f(Z), \ \text{non}linear\ activation\ \ function\ f : R \to R$$

Central difficulties of DNN

- Non-Convex
- High-Dimensional

RMT for the analysis of deep learning

Notation :

$X \in R^{n_0 \times m}$ $W \in R^{n_1 \times n_0}$ let non*linear activation funtion* $f : R \to R$ *with zero mean and finite moments*

W *and* X *are Gaussian ramdom matrices* with *i.i.d elements* $X_{i\mu} \sim N(0, \sigma_x^2), W_{ij} \sim N(0, \sigma_w^2 / n_0)$

Define $\phi \equiv \frac{n_0}{m}, \psi \equiv \frac{n_0}{n_1}$ *to be fixed cons*tan t*s*

Constants η *and* ς *defined as* :

$$\eta = \int \frac{e^{-z^2/2}}{\sqrt{2\pi}} f(\sigma_w \sigma_x z)^2 dz \qquad \varsigma = [\sigma_w \sigma_x \int \frac{e^{-z^2/2}}{\sqrt{2\pi}} f^{'}(\sigma_w \sigma_x z) dz]^2$$

RMT for the analysis of deep learning

Ramdom *Feature Map* :

$$Z = WX$$

$$Y = f(Z)$$

$$M = \frac{1}{m} YY^T \in R^{n_1 \times n_1}$$

Empirical Spectral Density :

$$\rho_M(t) = \frac{1}{n_1} \sum_{j=1}^{n_1} \delta(t - \lambda_j(M)),\ \delta \text{ is the Dirac delta function, } \lambda_j(M) \text{ denote the } j\text{th eigenvalue of } M$$

To analyze of the eigenvalues(eigenvalues distribution) of the Gram matrix M as it propagates through a neural network.

RMT for the analysis of deep learning

$$For\ z \in C \setminus \sup p(\rho_M)\ the\ Stieltjes\ transform\ G\ of\ \rho_M :$$

$$G(z) = \int \frac{\rho_M(t)}{z-t} dt = -\frac{1}{n_1} E[tr(M - zI_{n_1})^{-1}]$$

$$\rho_M(\lambda) = -\frac{1}{\pi} \lim_{\varepsilon \to 0^+} \operatorname{Im} G(\lambda + i\varepsilon)$$

But it's hard to solve the problem!

RMT for the analysis of deep learning

$$G(z)=\int\frac{\rho_M(t)}{z-t}dt=-\frac{1}{n_1}E[tr(M-zI_{n_1})^{-1}]$$

$$\rho_M(\lambda)=-\frac{1}{\pi}\lim_{\varepsilon\to 0^+}\operatorname{Im}G(\lambda+i\varepsilon)$$

Moment Method :

$$G(z)=\sum_{k=0}^{\infty}\frac{m_k}{z^{k+1}}\text{, } m_k \textit{ is the kth moment of the } \text{distribution } \rho_{\mathrm{M}}$$

$$\mathrm{m}_k=\int\rho_M(t)t^k dt=\frac{1}{n_1}E[trM^k]$$

$$\frac{1}{n_1}E[trM^k]=\frac{1}{n_1}E[\sum_{i_1,\dots,i_k\in[n_1]}M_{i_1i_2}M_{i_2i_3}\dots M_{i_{k-1}i_k}M_{i_ki_1}]$$

untraceable when n is infinite

RMT for the analysis of deep learning

the Stieltjes transform of the spectral density of M satisfies,

$$G(z) = \frac{\psi}{z} P\left(\frac{1}{z\psi}\right) + \frac{1-\psi}{z},$$

where,

$$P = 1 + (\eta - \zeta) t P_\phi P_\psi + \frac{P_\phi P_\psi t\zeta}{1 - P_\phi P_\psi t\zeta},$$

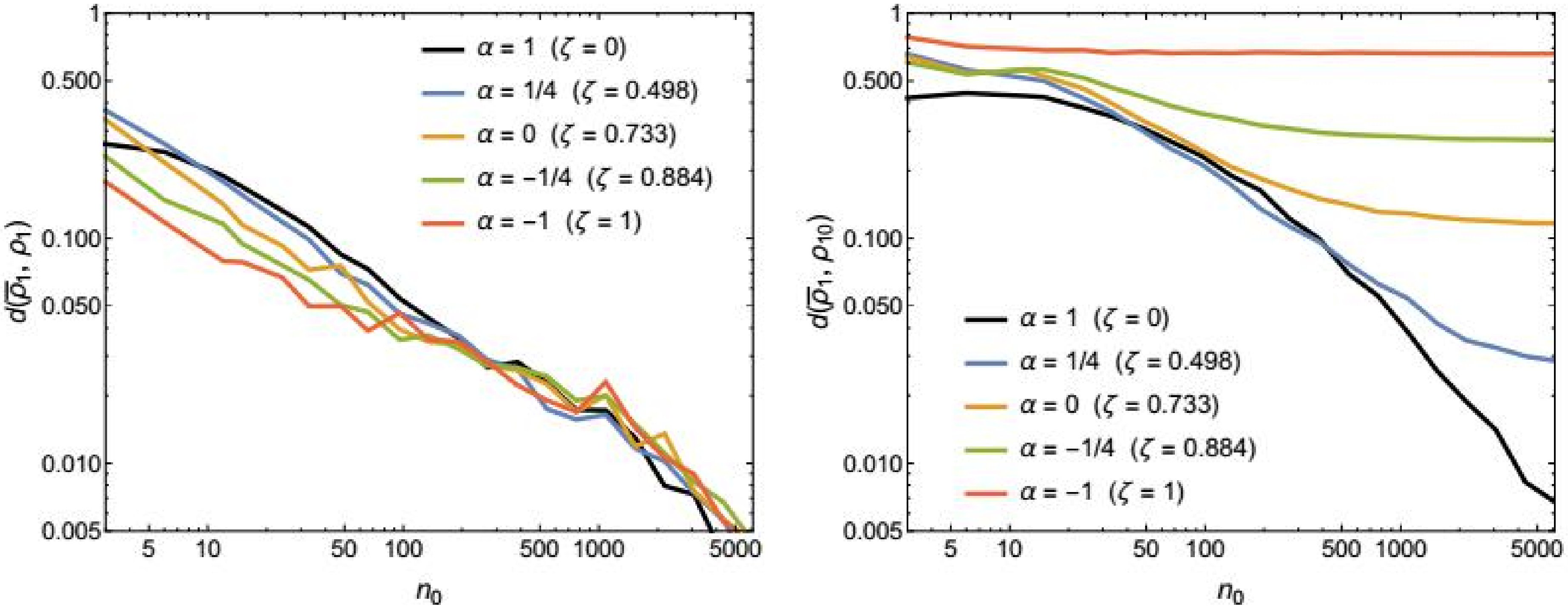
and

$$P_\phi = 1 + (P-1)\phi\,, \quad P_\psi = 1 + (P-1)\psi\,.$$

Another interesting limit is when $\zeta = 0$, which significantly simplifies the expression in eqn. (12). Without loss of generality, we can take $\eta = 1$ (the general case can be recovered by rescaling z). The resulting equation is,

$$z\,G^2 + \left(\left(1 - \frac{\psi}{\phi}\right)z - 1\right)G + \frac{\psi}{\phi} = 0\,, \tag{14}$$

which is precisely the equation satisfied by the Stieltjes transform of the Marchenko-Pastur distribution with shape parameter ϕ/ψ. Notice that when $\psi = 1$, the latter is the limiting spectral distribution of XX^T, which implies that YY^T and XX^T have the same limiting spectral distribution. Therefore we have identified a novel type of isospectral nonlinear transformation. We investigate this observation in Section 4.1.



(a) $L = 1$ (b) $L = 10$

Figure 1: Distance between the (a) first-layer and (b) tenth-layer empirical eigenvalue distributions of the data covariance matrices and our theoretical prediction for the first-layer limiting distribution $\bar{\rho}_1$, as a function of network width n_0. Plots are for shape parameters $\phi = 1$ and $\psi = 3/2$. The different curves correspond to different piecewise linear activation functions parameterize by α: $\alpha = -1$ is linear, $\alpha = 0$ is (shifted) relu, and $\alpha = 1$ is (shifted) absolute value. In (a), for all α, we see good convergence of the empirical distribution ρ_1 to our asymptotic prediction $\bar{\rho}_1$. In (b), in accordance with our conjecture, we find good agreement between $\bar{\rho}_1$ and the tenth-layer empirical distribution $\zeta = 0$, but not for other values of ζ. This provides evidence that when $\zeta = 0$ the eigenvalue distribution is preserved by the nonlinear transformations.

RMT for the analysis of deep learning

Highly skewed distributions indicate strong anisotropy in the embedded feature space, which is a form of poor conditioning that is likely to derail or impede learning.

Limitations

Strong Requirements:

- Activation function f : R→R with zero mean and finite moments
- Gaussian assumption
- $\varsigma = [\sigma_w \sigma_x \int \frac{e^{-z^2/2}}{\sqrt{2\pi}} f^{'}(\sigma_w \sigma_x z) dz]^2 = 0$

Hessian matrix of the loss function

$Decompose\ Hessian\ matrix$ at critical points int$o\ two\ \ pieces: H = H_0 + H_1$

w$here$

$$[H_0]_{\alpha\beta} \equiv \frac{1}{m}\sum_{i,\mu=1}^{n,m} \frac{\partial \hat{y}_{i,\mu}}{\partial \theta_\alpha} \frac{\partial \hat{y}_{i,\mu}}{\partial \theta_\beta} \equiv \frac{1}{m}[JJ^T]_{\alpha\beta}\ ,\ J\ is\ Jacobian\ matrix$$

$$[H_1]_{\alpha\beta} \equiv \frac{1}{m}\sum_{i,\mu=1}^{n,m} e_{i,\mu}\left(\frac{\partial^2 \hat{y}_{i,\mu}}{\partial \theta_\alpha \partial \theta_\beta}\right)$$

$$e_{i,\mu} = \hat{y}_{i,\mu} - y_{i,\mu}$$

Two assumptions

Primary assumptions:

1. The matrices H_0 and H_1 are *freely independent*, a property we discuss in sec. 3.
2. The residuals are i.i.d. normal random variables with tunable variance governed by ϵ, $e_{i\mu} \sim \mathcal{N}(0, 2\epsilon)$. This assumption allows the gradient to vanish in the large m limit, specifying our analysis to critical points.
3. The data features are i.i.d. normal random variables.
4. The weights are i.i.d. normal random variables.

Secondary assumption:

The elements of J and H_1 are i.i.d normal random variables.

RMT for the analysis of deep learning

Take $\sigma_{H_0} = 1$ and $\sigma_{H1} = \sqrt{2\varepsilon}$, where 2ε is the variance of $e_{i,\mu}, e_{i,\mu} \sim N(0, 2\varepsilon)$

then have $\rho_{H_0}(\lambda) = \rho_{MP}(\lambda; 1, \phi)$, $\rho_{H_1}(\lambda) = \rho_{SC}(\lambda; \sqrt{2\varepsilon}, \phi)$

where $\phi = 2n/m$

$$\rho_{\mathrm{MP}}(\lambda;\sigma,\phi) = \begin{cases} \rho(\lambda) & \text{if } \phi < 1 \\ (1-\phi^{-1})\delta(\lambda) + \rho(\lambda) & \text{otherwise} \end{cases} \qquad \rho_{\mathrm{SC}}(\lambda;\sigma,\phi) = \begin{cases} \frac{1}{2\pi\sigma^2}\sqrt{4\sigma^2-\lambda^2} & \text{if } |\lambda| \le 2\sigma \\ 0 & \text{otherwise} \end{cases}$$

where $\phi = n/p$ and,

$$\rho(\lambda) = \frac{1}{2\pi\lambda\sigma\phi}\sqrt{(\lambda-\lambda_-)(\lambda_+-\lambda)}$$

$$\lambda_\pm = \sigma(1 \pm \sqrt{\phi})^2 \,.$$

RMT for the analysis of deep learning

For :

$$G(z) = \int_R \frac{\rho(t)}{z-t} dt \quad (Stieltjes\ transform)$$

$$R(G(z)) + \frac{1}{G(z)} = z \quad (R\ transform)$$

$$R_{H_0+H_1} = R_{H_0} + R_{H_1} \ (H_0\ and\ H_1\ are\ freely\ independent)$$

We can have :

$$R_{H_0}(z) = \frac{1}{1-z\phi}, \, R_{H_1}(z) = 2\varepsilon z$$

$$R_H(z) = R_{H_0} + R_{H_1} = \frac{1}{1-z\phi} + 2\varepsilon z$$

$$2\varepsilon\phi G_H^3 - (2\varepsilon + z\phi) G_H^2 + (z+\phi-1) G_H - 1 = 0 \, , G_H \sim 1/z \ as\ z \to \infty \ (Tao, 2012)$$

RMT for the analysis of deep learning

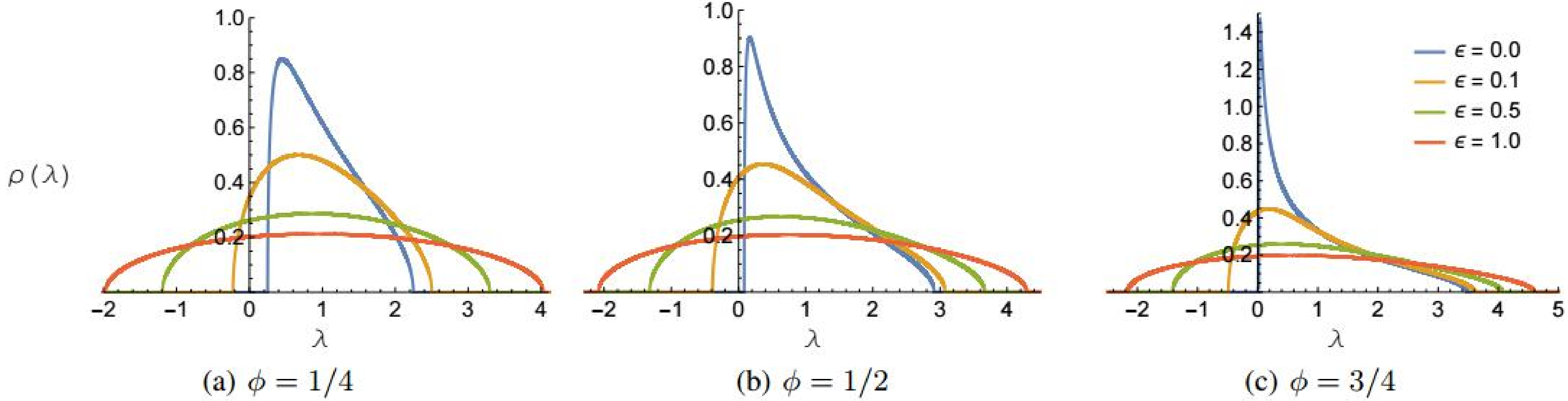


Figure 1. Spectral distributions of the Wishart + Wigner approximation of the Hessian for three different ratios of parameters to data points, ϕ. As the energy ϵ of the critical point increases, the spectrum becomes more semicircular and negative eigenvalues emerge.

RMT for the analysis of deep learning

What does that mean?

RMT for the analysis of deep learning

$$H = U\sum U^T = \sum \mathrm{u}_i u_i^T \sigma_i \text{ , } where\ (\sigma_i, u_i) \text{ are } eigenvalue\ and\ eigenvector$$

$$l(W) \approx l(W^*) + (W - W^*)^T \nabla l(W^*) + \frac{1}{2}(W - W^*)^T H(W^*)(W - W^*), W^*\ is\ a\ critical\ po\mathrm{int}$$

$$\approx l(W^*) + \frac{1}{2}(W - W^*)^T H(W^*)(W - W^*)$$

$$\approx l(W^*) + \frac{1}{2}(W - W^*)^T U_H \sum_H U_H^T (W - W^*)$$

$$\approx l(W^*) + \sum \sigma_{\mathrm{i}} \frac{1}{2}(W - W^*)^T u_i u_i^T (W - W^*)$$

$$\Rightarrow$$

$$l(W) - l(W^*) \approx \sum \sigma_{\mathrm{i}} \frac{1}{2}(W - W^*)^T u_i u_i^T (W - W^*)$$

RMT for the analysis of deep learning

So we can conclude:

- As the energy ε of the critical point increase, the critical point's probability of being saddle point will increase!
- Increase the ratio Φ(#parameters/#samples), it could be more probably to escape from the saddle points.

References

- Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks[J]. 2014, 4:3104-3112.
- Sordoni A, Bengio Y, Vahabi H, et al. A Hierarchical Recurrent Encoder-Decoder for Generative Context-Aware Query Suggestion[C]// ACM International on Conference on Information and Knowledge Management. ACM, 2015:553-562.
- Pennington J, Worah P. Nonlinear random matrix theory for deep learning[C]//Advances in Neural Information Processing Systems. 2017: 2637-2646.
- Pennington J, Bahri Y. Geometry of neural network loss surfaces via random matrix theory[C]//International Conference on Machine Learning. 2017: 2798-2806.
- Tao, Terence. Topics in random matrix theory, volume 132. American Mathematical Society Providence, RI, 2012.
- https://zhenyu-liao.github.io/pdf/tutorial_Eusipco_handout.pdf
- https://zhuanlan.zhihu.com/p/38777140

谢　谢！