

Title: PSI 2017/2018 - Machine Learning for Many Body Physics - Lecture 9

Date: Apr 19, 2018 09:00 AM

URL: <http://pirsa.org/18040062>

Abstract:

Unsupervised learning

* Clustering $\begin{cases} \rightarrow \text{PCA} \\ \rightarrow \text{t-SNE} \end{cases}$

X Density estimation: given $\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N \rightarrow \text{infer } P_{\text{data}}(\vec{x})$

$P_{\Theta}(\vec{x}, \vec{h}) \rightarrow P_{\Theta}(\vec{x}) = \sum_{\vec{h}} P_{\Theta}(\vec{x}, \vec{h})$, ex: Restricted Boltzmann Machine (RBM)

Unsupervised learning

* Clustering $\begin{cases} \rightarrow \text{PCA} \\ \rightarrow \text{t-SNE} \end{cases}$

X Density estimation: given $\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N \rightarrow \text{infer } P_{\text{data}}(\vec{x})$

• $P_{\Theta}(\vec{x}, \vec{h}) \rightarrow P_{\Theta}(\vec{x}) = \sum_{\vec{h}} P_{\Theta}(\vec{x}, \vec{h})$, ex. Restricted Boltzmann Machine (RBM)

• $P_{\Theta}(\vec{x} | \vec{h}), P(\vec{h}) \quad P_{\Theta}(\vec{x}) = \sum_{\vec{h}} P_{\Theta}(\vec{x} | \vec{h}) P(\vec{h}) \quad (\text{latent variable models})$

Unsupervised learning

* Clustering $\begin{cases} \rightarrow \text{PCA} \\ \rightarrow \text{t-SNE} \end{cases}$

x Density estimation: given $\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N \rightarrow \text{infer } P_{\text{data}}(\vec{x})$

• $P_{\Theta}(\vec{x}, \vec{h}) \rightarrow P_{\Theta}(\vec{x}) = \sum_{\vec{h}} P_{\Theta}(\vec{x}, \vec{h})$, ex. Restricted Boltzmann Machine (RBM)

• $P_{\Theta}(\vec{x} | \vec{h}), P(\vec{h}) \quad P_{\Theta}(\vec{x}) = \sum_{\vec{h}} P_{\Theta}(\vec{x} | \vec{h}) P(\vec{h}) \quad (\text{latent variable models})$

• $P_{\Theta}(x_1) P_{\Theta}(x_2 | x_1) P_{\Theta}(x_3 | x_1, x_2) \dots = P_{\Theta}(\vec{x})$

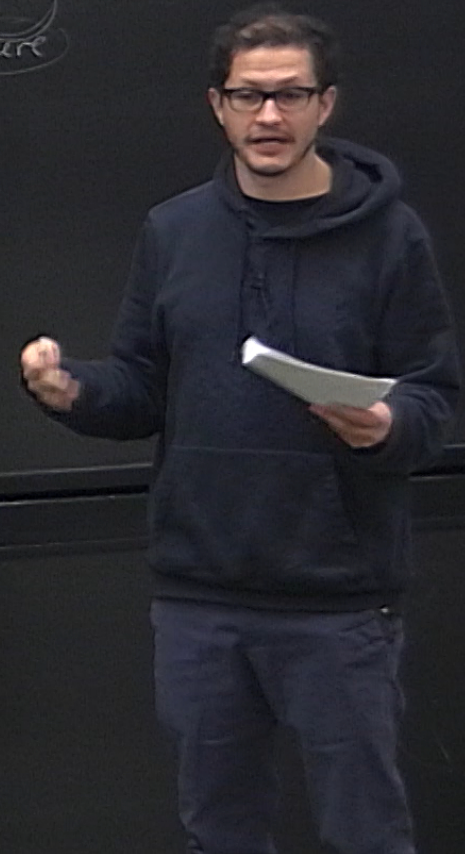
$P_\theta(\vec{x}) = \prod_{t=1}^n P(x_t | x_{<t})$ is a natural decomposition for problems which involve sequences

x Natural language. The cats, after eating carrots once , was full
were

x 1D spin systems $\uparrow$
 $\downarrow$

x time evolution of quantum system.

$\oplus$



- $P_0(x|h), P(h)$ $P_0(\vec{x}) = \sum_h P_0(\vec{x}|h) P(h)$ (latent variable models)
- $P_0(x_1) P_0(x_2|x_1) P_0(x_3|x_1, x_2) \dots = P_0(\vec{x})$

Recurrent neural network probabilistic models (RNN)

- + similar to NN but the weights between time steps in a sequence remain the same.
- + they allow us to parametrize $P_0(\vec{x})$ using chain rule.

- $P_0(x|h), P(h)$ $P_0(\vec{x}) = \sum_h P_0(\vec{x}|h) P(h)$ (latent variable models)
- $P_0(x_1) P_0(x_2|x_1) P_0(x_3|x_1, x_2) \dots = P_0(\vec{x})$

Recurrent neural network probabilistic models (RNN)

- + similar to NN but the weights between time steps in a sequence remain the same.
- + they allow us to parametrize $P_0(\vec{x})$ using chain rule.

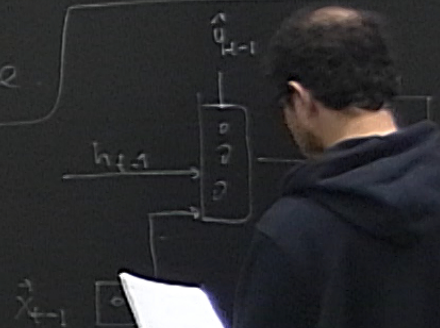
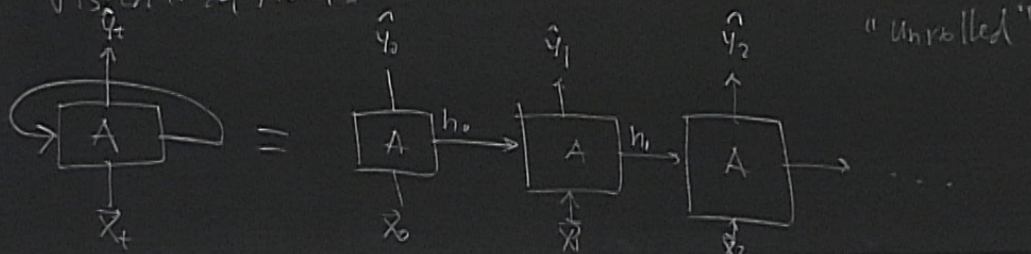
Visualization:

$$P_0(x_1) P_0(x_2|x_1) P_0(x_3|x_1, x_2) \dots = P_0(\vec{x})$$

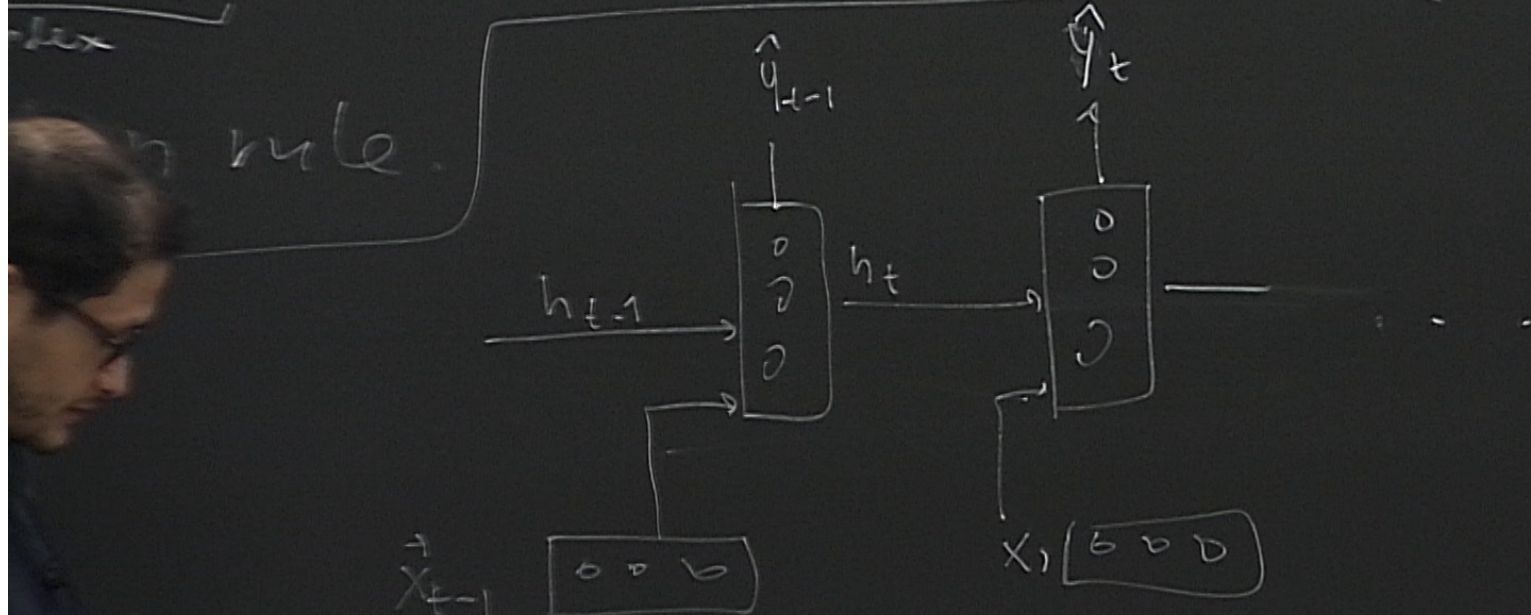
Recurrent neural network probabilistic models (RNN)

- + similar to NN but the weights between time steps index in a sequence remain the same.
- + they allow us to parametrize $P_0(\vec{x})$ using chain rule.

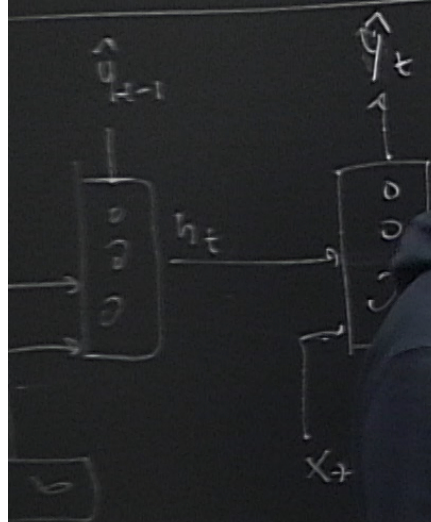
Visualization:



steps in a sequence remain the same.



ce remain the same.



Simplest RNN

$$h_t = f_h (W h_{t-1} + V x_t)$$

$$\hat{y}_t = f_y (U h_t)$$

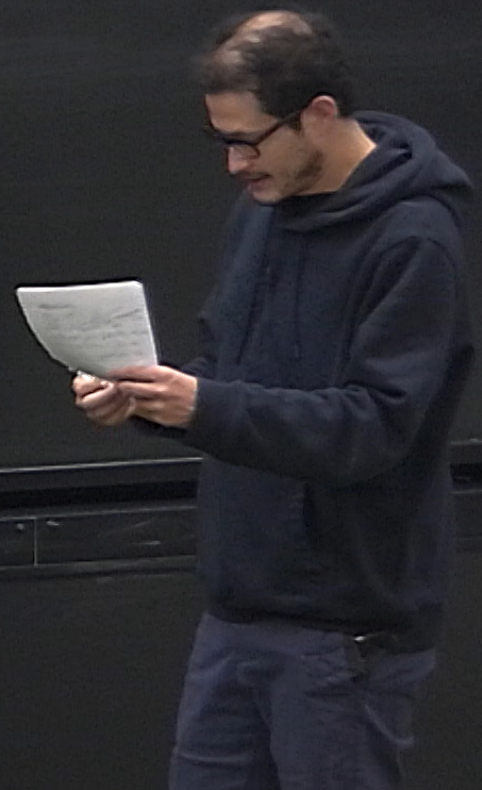
Softmax, $\hat{q} = (q_1, \dots, q_m)$

$$\text{Softmax}(q_i) = \frac{e^{q_i}}{\sum_j e^{q_j}}$$

In the probabilistic model

* Use a softmax

* Assign $P(X_{t+1} | X_t, X_{t-1}, \dots, X_1) \rightarrow \hat{y}_t$



In the probabilistic model

* Use a softmax

* assign $P(X_{t+1} | X_t, X_{t-1}, \dots, X_1) \rightarrow \hat{y}_t$

Cost function = cross-entropy loss

$$L_t = - \sum_{j=1}^V y_{t,j} \ln \hat{y}_{t,j}$$

$$L = \sum_t L_t$$

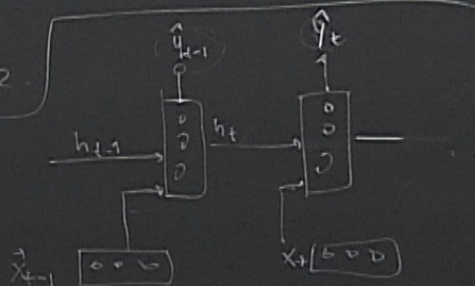
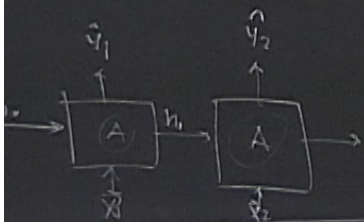
$$= P_\theta(z)$$

probabilistic models (RNN)

but the weights between time steps in a sequence remain the same.

parametrize $P_\theta(x)$ using chain rule.

"Unrolled"



Simplest RNN

$$h_t = f_h(W h_{t-1} + V x_t)$$

$$\hat{y}_t = f_y(U h_t)$$

Softmax, $\hat{q} = (q_1, \dots, q_n)$

$$\text{Softmax}(q_i) = \frac{e^{q_i}}{\sum_j e^{q_j}}$$

$(x_1, \dots, x_n) \rightarrow$

$\hat{y}_t$

entropy loss

$$\frac{\partial L}{\partial u} = \sum_t \frac{\partial L_t}{\partial u} = \sum_t \frac{\partial L}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial u}$$

$$\frac{\partial L_t}{\partial w} = \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_t} \left(\frac{\partial h_t}{\partial w} \right) !! ?$$

total derivative

$$y = f(t, u_1(t), u_2(t), \dots, u_m(t))$$

$$\frac{\partial y}{\partial t} = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial u_1} \frac{\partial u_1}{\partial t} + \frac{\partial f}{\partial u_2} \frac{\partial u_2}{\partial t} + \dots$$

$$\frac{\partial L}{\partial y} = \sum_t \frac{\partial L_t}{\partial y} = \sum_t \frac{\partial L}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial y}$$

$$\frac{\partial L_t}{\partial w} = \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_t} \left(\frac{\partial h_t}{\partial w} \right) \quad !!?$$

total derivative

$$y = f(t, u_1(t), u_2(t), \dots, u_m(t))$$

$$\frac{\partial y}{\partial t} = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial u_1} \frac{\partial u_1}{\partial t} + \frac{\partial f}{\partial u_2} \frac{\partial u_2}{\partial t} + \dots$$

$$\frac{\partial L_t}{\partial w} = \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_t} \sum_{k=0}^t \left(\prod_{i=k+1}^t \frac{\partial h_i}{\partial h_{i-1}} \right) \frac{\partial h_k}{\partial w}$$

α^t

$$\alpha < 1.0$$

$$\alpha > 1.0$$

$$\frac{\partial L}{\partial y} = \sum_t \frac{\partial L_t}{\partial y} = \sum_t \frac{\partial L}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial y}$$

$$\frac{\partial L_t}{\partial w} = \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_t} \left(\frac{\partial h_t}{\partial w} \right) !!!$$

total derivative

$$y = f(t, u_1(t), u_2(t), \dots, u_m(t))$$

$$\frac{\partial y}{\partial t} = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial u_1} \frac{\partial u_1}{\partial t} + \frac{\partial f}{\partial u_2} \frac{\partial u_2}{\partial t} + \dots$$

$$\frac{\partial L_t}{\partial w} = \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_t} \sum_{k=0}^t \left(\prod_{i=k+1}^t \frac{\partial h_i}{\partial h_{i-1}} \right) \frac{\partial h_k}{\partial w}$$

www.deeplearningbook.org

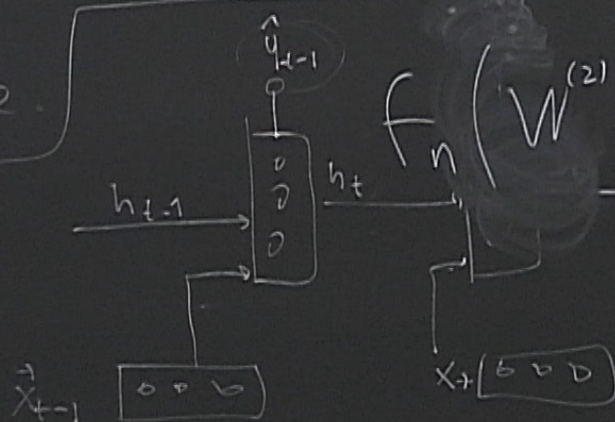
10.2.2 section

NN)

time steps in a sequence remain the same.

index

chain rule.



- LSTMs : Long short term memory

- GRU : Gated recurrent units

$$f_n \left(W^{(2)} f_n \left(W^{(1)} f_n (W h_{t-1} + V x_t) \right) \right)$$

$$p_0(x_1) p_0(x_2|x_1) p_0(x_3|x_1, x_2) \dots = p_0(\vec{x})$$

Recurrent neural network probabilistic models (RNN)

- + similar to NN but the weights between time steps in a sequence remain the same.
- + they allow us to parametrize $p_0(\vec{x})$ using chain rule.

Visualization:

