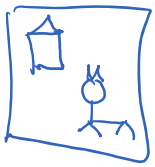
Lecture 22 - latent SVMstruct; RNN

Thursday, April 6, 2023 1:32 PM

today: · latent variable SVMstruct – CCCP
· deep learning – RNN

latent variables

motivation: semantic segmentation



segmentation is expensive → z "latent variable"

perhaps only have class labels → y

also: [Felzenszwalb et al. TPAMI 2010]

"deformable part models" for object recognition
↳ z there was an object part configuration

before, we had $s(x,y;w) = \langle w, \varphi(x,y)\rangle$

now, consider $s(x,y,z;w) = \langle w, \varphi(x,y,z)\rangle$

as before, could predict with $\underset{y\in\mathcal{Y}, z\in\mathcal{Z}}{\operatorname{argmax}} s(x,y,z;w) \triangleq h_w(x)$

learning
- CRF approach $p(y|x)$ — generalize → hidden CRF $p(y,z|x)$
  - similarly to latent variable modeling with PGM
  - → marginalize z out
  - ML → EM (expectation-maximization)
  - analog for latent SVMstruct is CCCP (majorization-minimization procedure → like EM)
- SVMstruct (surrogate loss minimization)

latent SVMstruct

$\ell(y,(\tilde{y},\tilde{z}))$

generalize structure hinge loss:

$$g(x,y;w) \triangleq \max_{\tilde{y},\tilde{z}} \langle w, \varphi(x,\tilde{y},\tilde{z})\rangle + \ell(y,(\tilde{y},\tilde{z})) - \max_{z'\in\mathcal{Z}} \langle w, \varphi(x,y,z')\rangle \geq \ell(y, h_w(x))$$

best score for ground truth

$$f(x,y;w) \triangleq \underbrace{\max_{\tilde{y},\tilde{z}} \langle w, \varphi(x,\tilde{y},\tilde{z})\rangle + \ell(y,(\tilde{y},\tilde{z}))}_{\triangleq u(w)} - \underbrace{\max_{z' \in Z} \langle w, \varphi(x,y,z')\rangle}_{\triangleq v(w)} \geq \ell(y, h_w(x))$$

where $f(x,y;w) = u(w) - v(w)$ where u & v are convex fct. of w

"difference of convex functions" (dc)

↳ CCCP procedure is to approx. minimize this

CCCP procedure:

- linearize $v(w)$ at w_t to get an upper bound
- w_{t+1} is obtained by minimizing upper bound
- repeat ↳ a majorization–minimization procedure (EM is another example)

$$\begin{cases} f_t(w) = u(w) - [v(w_t) + \langle \nabla v(w_t), w - w_t\rangle] \geq f(w) \quad \forall w \\ w_{t+1} = \operatorname{argmin}_w f_t(w) \end{cases}$$

(↑ or subgradient) and $f_t(w_t) = f(w_t)$

properties of procedure: • like EM, descent procedure i.e. $f(w_{t+1}) \leq f(w_t)$

$$f(w_t) = f_t(w_t) \geq f_t(w_{t+1}) \geq f(w_{t+1}) \quad //$$

(majorization property)

- local linear convergence to a stationarity pt. for latent SVMstruct

[see NeurIPS OPT 2012 paper]

---

* **CCCP for SVMstruct:** $v(w) = \max_{z'} \langle w, \varphi(x,y,z')\rangle$

$$\partial v(w_t) = \varphi(x,y,\hat{z}(x,y;w_t)) \quad \text{where } \hat{z} \triangleq \operatorname{argmax}_{z'} \langle w_t, \varphi(x,y,z')\rangle$$

(project idea: do amortization with NN $\varphi(x,y) \to \hat{z}$)

$$\Rightarrow f_t(w) = \max_{(\tilde{y},\tilde{z})} \langle w, \varphi(x,\tilde{y},\tilde{z})\rangle + \ell(y,(\tilde{y},\tilde{z})) - \langle w, \varphi(x,y,\hat{z}_t)\rangle + \text{cst.}$$

↝ like SVMstruct obj.

CCCP alg. for latent SVM struct

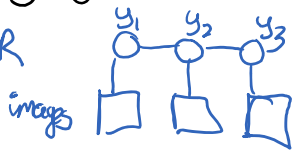
repeat:
- fill in $\hat{z}_t^{(i)}$ for all ground truth $y^{(i)}$ using w_t
- solve a standard SVM struct to get w_{t+1}
- repeat

Deep learning:

go from $\langle w, \varphi(x,y) \rangle$ to $\langle w, \varphi(x,y;\theta) \rangle$ (can learn)

I) plug-in "deep learning" features in a structured prediction model

examples: OCR

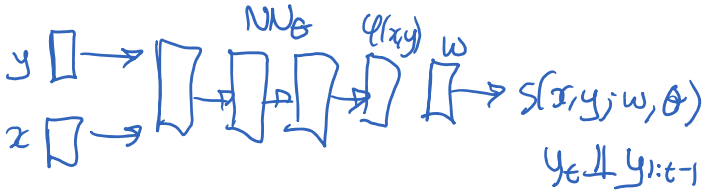


example: [Vu & al. ICCV 2015] "context aware CNNs for" person-head detection

so far $\varphi_t(x_t, y_t) = \begin{pmatrix} 0 \\ x_t \\ 0 \end{pmatrix}$ ← y_t^{th} position

instead $\varphi_t(x_t, y_t) = \begin{pmatrix} 0 \\ NN_\theta(x_t) \\ 0 \end{pmatrix}$ ← y_t^{th} position; NN_θ pre-trained on imagenet e.g.

II) "end-to-end" training — structured prediction energy networks (SPENs)



14h28

III) reccurent neural networks (RNN)

motivation: $p(y|x) \overset{\text{chain rule}}{=} \prod_{t \in T} p(y_t | y_{1:t-1}, x)$

graphical modelling approach:
- $\prod_t p(y_t | y_{\pi_t}, x)$ [directed]
- $\frac{1}{Z(x)} \prod_{c \in \mathcal{C}} \psi_c(y_c; x)$ [undirected]

$y_t \perp y_{1:t-1} | y_{\pi_t}$

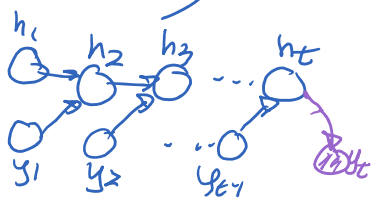
RNN → "structured parameterization" of $p(y_t | y_{1:t-1}, x)$ using NN

with no cond. indep assumptions → usually loose exact decoding

$h_{t+1} \triangleq f(h_t, x, y_t, W)$

$h_t = f(f(\cdots f(h_1, x, y_1, w), x, y_1, w) \cdots, x, y_{t-1}, w)$

define $p(y_t \mid y_{1:t-1}, x) \propto \exp(c(y_t)^\top \tilde{w} h_t)$



"code"

e.g. word embedding in MT and can be fine tuned

$p(y_t \mid y_{1:t-1}, x)$ given by a deep NN architecture

standard learning: using ML

i.e. $\min_{w, \tilde{w}} -\frac{1}{n} \sum_{i=1}^{n} \log p(y^{(i)} \mid x^{(i)})$

$\sum_t \log p(y_t^{(i)} \mid y_{1:t-1}^{(i)}, x^{(i)})$

output of a deep NN

"teacher forcing"

"exposure problem" ie. don't know $p(\tilde{y}_t \mid \text{unseen } \tilde{y}_{1:t-1}, x)$

for ML, do SGD on objective

gradient of $\log p(y_t^{(i)} \mid y_{1:t-1}^{(i)}, x^{(i)}; w, \tilde{w})$ → use back propagation

decoding: $\arg\max_{y \in \mathcal{Y}} \sum_t \log p(y_t \mid y_{1:t-1}, x)$ → NP hard?

→ need approximation

- greedy decoding: $\hat{y}_t = \arg\max_{y_t \in \mathcal{Y}_t} p(\tilde{y}_t(\hat{y}_{1:t-1}, x))$
- beam search: "greedy decoding with memory of ~~size~~ K" — K = size of beam

beam search construct $\hat{y}_1, \ldots, \hat{y}_T$

beam of size L (memory)

- at step t, you have L candidate solution prefixes $y_{1:t}^{(1)}, \ldots, y_{1:t}^{(L)}$
- expand possible next choices $|\mathcal{Y}_{t+1}| \cdot L$

score them (e.g. $\log p(y_{t+1} \mid \hat{y}_{1:t}^{(e)}, x) + \log p(\hat{y}_{1:t}^{(e)}, x)$)

then keep top L candidates as $\{\hat{y}_{1:t+1}^{(e)}\}_{e=1}^{L}$

score : $(\text{e.g. } \log p(y_{t+1} \mid y_{1:t}, x) + \log p(y_{1:t}, x))$

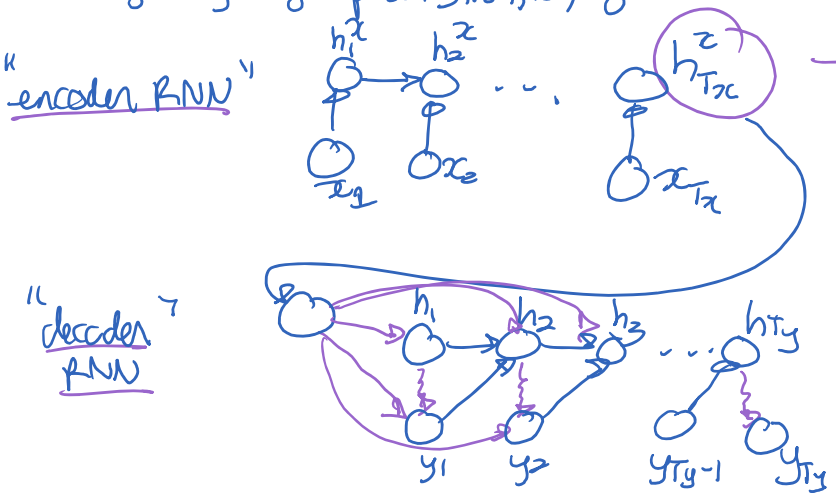
then keep top L candidates as $\{\hat{y}^{(l)}_{1:t+1}\}_{l=1}^{L}$

vs.

Viterbi alg. which does "backtracking" to correct past mistakes

seq2seq aka. encoder/decoder architecture

↳ useful way to get $p(y_t \mid y_{1:t-1}, x)$ for a RNN when x has variable length



issues:

a) variable length output? → end-of-sequence character

b) how to handle long sequence x?

problem: need to summarize input sentence in one context vector of fixed length

solution: "attention mechanism"

c) vanishing gradient?

- LSTM
- gated recurrent unit (GRU)
- etc.

d) sequential processing

→ fixed by "transformer architecture"

→ GPT & cl.