10-708: Probabilistic Graphical Models 10-708, Spring 2018

22 : A Hybrid of Deep Learning and Graphical Models

Lecturer: Kayhan Batmanghelich, Mingming Gong *Scribes: Shaojie Bai, Chih-Kuan Yeh*

1 Recap: VAE

VAE learns a mapping that transforms a simple distribution, $q(z)$, to a complicate distribution, $q(z|x)$ so that the empirical distribution (in the x-space) are matched. However, the main limitations of VAE is that it results in a blurry image, as well as its difficulty to apply to discrete latent variables.

2 Generative Adversarial Networks

Generative Adversarial Networks (GAN) assumes a implicit generative model without Markov Chains assumption, which result in the best samples to human eyes. The procedure for GAN is to map noise to the data distribution, where it trains a generator and a discriminator alternatively. The goal for the generator is to generate samples which fools the discriminator, while the discriminator tries to identify the samples from true data distribution and samples generated by the generator. The final objective function becomes:

$$\min_G \max_D \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}[log(D(\mathbf{x}))] + \mathbb{E}_{\mathbf{z}\sim p_{\text{data}}(\mathbf{z})}[log(1 - G(D(\mathbf{z})))], \tag{1}$$

which can be seen as a minimax game between two players, the generator and the discriminator.

An interesting fact is that GAN actually minimizes the JensenShannon divergence between two distributions. In practice, first proposed in DC-GAN, adding batch normalization effectively prevents the mode collapse problem for training GANs.

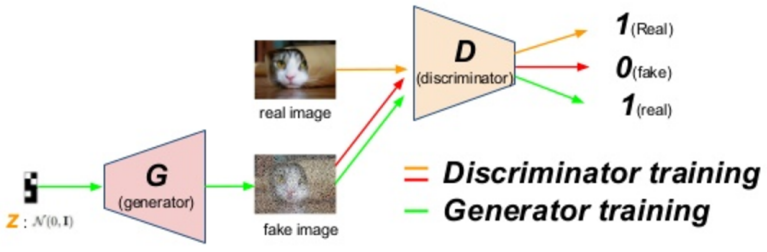


Figure 1: Illustration of training GANS

2.1 Summary for GANs

- Does not need a specific likelihood function at the last layer to train a latent-variable model
- Uses a neural network to inference instead of restrictions

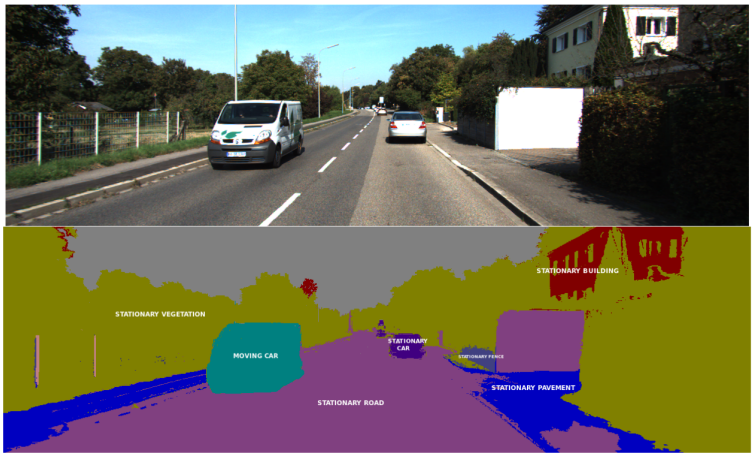
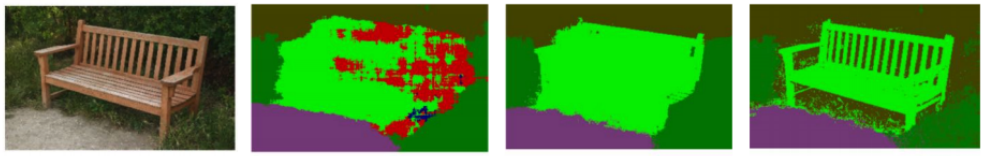


Figure 2: Illustration of semantic segmentation



(a) Image (b) Unary classifiers (c) Robust P^n CRF (d) Fully connected CRF, MCMC inference. 36 hrs

Figure 3: An example of doing semantic segmentation on a bench, using different methods

- Generates images that is most realistic (though no way to evaluate)

3 Application in Vision - Semantic Segmentation

Graphical models also have plenty of application in the field of computer vision. Once such example is semantic segmentation, and another usage is in image-to-image translation (next class).

3.1 Semantic Segmentation

In a semantic segmentation problem, we want to partition an visual sample (image) into semantically meaningful regions. For instance, in Figure 2, each object in the background (vans, cars, pavement, etc.) is highlighted with a different color with clear boundaries:

We therefore are performing a *pixel-wise classification*. Essentially, we want to predict a label for each pixel of the image, and ideally nearby pixels should be highly related, which means they have a higher chance of having the same label.

Figure 3 shows an example where we apply different modeling of the pixel relations to the segmentation. Using unary classifiers, we don't assume explicit interactions between pixels, resulting in a bad model that is not able to tell the background of the bench. With robust P^n classifier (grid CRF), we can do better, but a much better result was obtained with *fully connected CRF + MCMC inference* (see Figure 4).

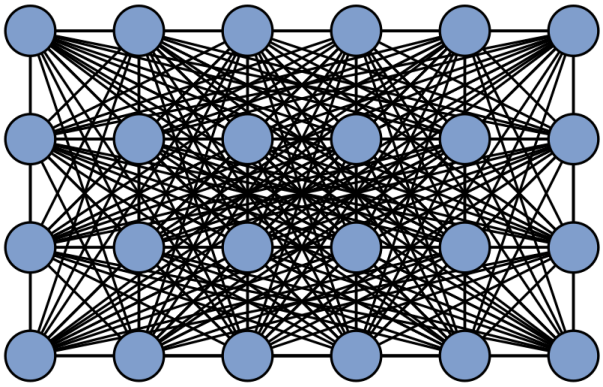


Figure 4: Fully Connected CRF

However, a critical issue with fully connected CRF is, it also models long-range connections (which we probably don't need). More importantly, complexity of inference in such fully connected models has restricted their application to sets of hundreds of image regions or fewer [3].

A fully connected CRF has an energy function of the form

$$E(x) = \sum_i \psi_u(x_i) + \sum_{i<j} \underbrace{\psi_p(x_i, x_j)}_{\text{models all pairs}}$$

where we can clearly see a unary potential term and a pairwise potential term. The pairwise potential, more specifically, takes the form:

$$\psi_p(x_i, x_j) = \mu(x_i, x_j) \sum_{m=1}^{K} w^{(m)} k^{(m)}(f_i, f_j)$$

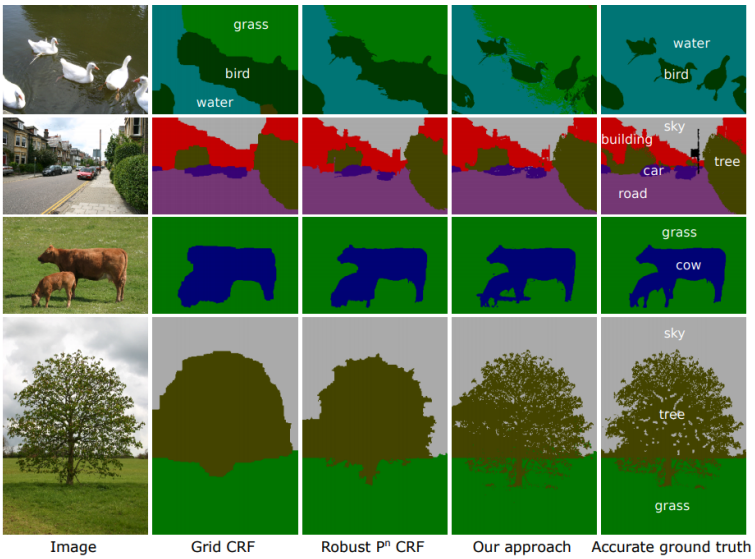
μ is called the label compatibility, which is 0 if we have the same label for x_i and x_j and a positive penalty if otherwise. k here is kernel function that can be decomposed into two parts:

$$k(f_i, f_j) = w^{(1)} \underbrace{\exp\left(-\frac{|p_i - p_j|^2}{2\theta_\alpha^2} - \frac{|I_i - I_j|^2}{2\theta_\beta^2}\right)}_{\text{appearance kernel}} + w^{(2)} \underbrace{\exp\left(-\frac{|p_i = p_j|^2}{2\theta_\gamma^2}\right)}_{\text{smoothness kernel}}$$

where $w^{(1)}$ measures a similarity between feature/pixel locations, and $w^{(2)}$ measure the similarity between features themselves. In a simpler way to describe the appearance kernel, we only compare two features if they are close enough in some range.

3.2 High-dimensional Filtering

For inference in a full CRF, one can find the most likely assignment (MAP): $\hat{x} = \text{argmax}_x P(x)$ where $P(x) = \exp(-E(x))$. But P can be hard to compute, and a usual solution is to perform a mean-field approximation and find $Q(x) = \prod_i Q(x_i)$ that is close to P in KL measure.



	Runtime	Standard ground truth		Accurate ground truth	
		Global	Average	Global	Average
Unary classifiers	–	84.0	76.6	83.2 ± 1.5	80.6 ± 2.3
Grid CRF	$1s$	84.6	77.2	84.8 ± 1.5	82.4 ± 1.8
Robust P^n CRF	$30s$	84.9	77.5	86.5 ± 1.0	83.1 ± 1.5
Fully connected CRF	**0.2s**	**86.0**	**78.3**	**88.2** ± 0.7	**84.7** ± 0.7

Figure 5: Fully Connected CRF on the MSRC-21 dataset

It turns out

$$Q_i(x_i = \ell) = \frac{1}{Z_i} \exp\left[-\psi_u(x_i) - \sum_{\ell' \in \mathcal{L}} \mu(\ell, \ell') \sum_{m=1}^{K} w^{(m)} \underbrace{\sum_{j \neq i} k^{(m)}(f_i, f_j) Q_j(\ell')}_{\tilde{Q}_i^{(m)}(\ell)} \right]$$

The mean field approximation involves 5 steps (initialization, message passing, compatibility transform, local update, and normalization) [3]. In the message passing phase, we can use *high-dimensional filtering* [1].

In particular, we would like to update all $\tilde{Q}_i^{(m)}(\ell)$ simultaneously:

$$\tilde{Q}_i^{(m)}(\ell) = \sum_{j \neq i} k^{(m)}(f_i, f_j) Q_j(\ell')$$

This can be efficiently computed using a cross-bilateral filter. The idea is to instead compute

$$\bar{Q}_i^{(m)}(\ell) = \sum_{j \in \sqsubseteq} \exp\left(\frac{1}{2} (f_i^{(m)} - f_j^{(m)})^2 \right) Q_j(\ell)$$

which is equivalent to a convolution operator: $\bar{Q}_i^{(m)}(\ell) = [G_{\Gamma^{(m)}} \otimes Q(\ell)](f_i) - Q_i(\ell)$.

Applying this result, [3] showed that it significantly outperformed other methods and is extremely close to the groundtruth (see Figure 5 for results). This approach also yields very good result on other famous vision datasets, such as PASCAL VOC 2010.

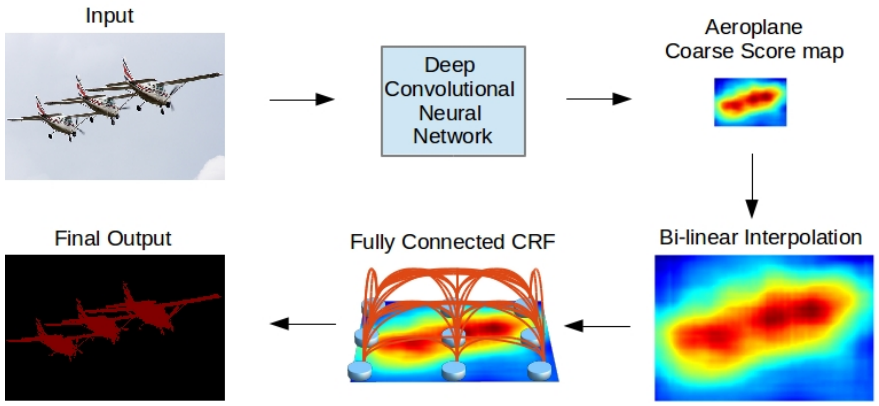


Figure 6: Using deep CNN along with CRF for better feature extraction

3.3 Convolutional Neural Nets

One limitation with CRF is that the unary classifiers are trained on handcrafted features. A better way, thanks to the development of deep networks, is to use deep ConvNets for dense feature extraction. The process is very straightforward (see Figure 6), as we simply perform fully connected CRF on extracted/processed features obtained from a deep convolutional network. An even better way is to expand the receptive field of the ConvNet used using atrous/dilated convolutions [4]. Relevant works [2] have shown that CNN-based methods significantly outperforms other previous methods that rely on handcrafted features.

References

[1] Andrew Adams, Jongmin Baek, and Myers Abraham Davis. Fast high-dimensional filtering using the permutohedral lattice. In *Computer Graphics Forum*, volume 29, pages 753–762. Wiley Online Library, 2010.

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018.

[3] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pages 109–117, 2011.

[4] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *International Conference on Learning Representations*, 2016.