Implicit Training of Inference Network Models for Structured Prediction (Supplementary Material)

1 IMPLICIT GRADIENT

Consider the following bi-level objective

$$\phi^* = \operatorname*{argmin}_{\phi} \mathcal{L}_{\text{Prim}}(\theta * (\phi), \phi) \text{ s.t. } \theta^*(\phi) = \operatorname*{argmin}_{\theta} \mathcal{L}_{\text{Aux}}(\theta, \phi) \tag{1}$$

Here we have explicitly added the dependence of θ due to the opimization process on ϕ. One approach to find the optimal ϕ^* is to find the partial derivative of $\mathcal{L}(f(\theta^*(\phi)), D_{\text{val}})$ with respect to the ϕ, and use gradient descent based optimization. The corresponding partial derivative is given by

$$\frac{d}{d\phi}\mathcal{L}_{\text{Prim}}(\theta^*(\phi), \phi) = \underbrace{\frac{\partial}{\partial\theta}(\mathcal{L}_{\text{Prim}}(\theta^*(\phi), \phi))|_{\theta^*}}_{1} \circ \underbrace{\frac{\partial}{\partial\phi}\theta^*}_{II} + \frac{\partial}{\partial\phi}\mathcal{L}_{\text{Prim}}(\theta^*(\phi), \phi) \tag{2}$$

Considering that the inner optimization is finished we have direct access to $\theta^*(\phi)$ and the first term I in the previous equation can be computed directly. The second term is a more challenging to compute.

Implicit gradient method computes this gradient via differentiation of the optimality criteria of the inner optimization. The optimality criteria states that the gradient of the inner loss at the optima $\theta^*(\phi)$ is zero i.e.

$$\nabla_\theta \mathcal{L}_{\text{Aux}}(\theta, \phi) = 0 \Rightarrow \frac{\partial}{\partial\theta}(\mathcal{L}_{\text{Aux}}(\theta, \phi)) = 0 \tag{3}$$

By differentiating this with respect to ϕ one gets:

$$\frac{d}{d\phi}\frac{\partial}{\partial\theta}\mathcal{L}_{\text{Aux}}(\theta, \phi) = 0 \Rightarrow \frac{\partial}{\partial\phi}\frac{\partial}{\partial\theta}(\mathcal{L}_{\text{Aux}}(\theta, \phi)) + \frac{\partial}{\partial\theta}\frac{\partial}{\partial\theta}(\mathcal{L}_{\text{Aux}}(\theta, \phi))\frac{\partial}{\partial\phi}\theta = 0 \tag{4}$$

$$\rightarrow \frac{\partial\theta}{\partial\phi} = -\left[\frac{\partial}{\partial\phi}\frac{\partial}{\partial\theta}(\mathcal{L}_{\text{Aux}}(\theta, \phi))\right]\left[\frac{\partial}{\partial\theta}\frac{\partial}{\partial\theta}(\mathcal{L}_{\text{Aux}}(\theta, \phi))\right]^{-1} \tag{5}$$

Putting this back in Equation 2 we get

$$\frac{d}{d\phi}\mathcal{L}_{\text{Prim}}(\theta^*(\phi), \phi) = -\left[\frac{\partial}{\partial\phi}\frac{\partial}{\partial\theta}(\mathcal{L}_{\text{Aux}}(\theta, \phi))\right]\left[\frac{\partial}{\partial\theta}\frac{\partial}{\partial\theta}(\mathcal{L}_{\text{Aux}}(\theta, \phi))\right]^{-1}\frac{\partial(\mathcal{L}_{\text{Prim}}(\theta^*(\phi), \phi)}{\partial\theta} + \frac{\partial(\mathcal{L}_{\text{Prim}}(\theta^*(\phi), \phi))}{\partial\phi} \tag{6}$$

Remark 1. *Notice that in Equation 6, the inverse of second order derivative i.e. a Hessian needs to be computed which can be expensive. In practice, approximations via Conjugate Gradient, Shermon-Morrison Identity, diagonalized Hessian or von-Neumann Expansion could be used. For our structure prediction experiments we used the von-Neumann approximation.*

Dataset	Train	Valid	Test	Label
Bibtex	4407	1491	1497	159
Delicious	9690	3207	3194	983
Eurlexev	11557	3876	3881	3993

Table 1: Summary statistics for multi-label classification datasets

Dataset	Size	Label	Avg Length	Avg Labels
AAPD	55840	54	163.4	2.4
RCV	804414	103	123.9	3.2

Table 2: Summary statistics for large classification datasets

Table 3: Summary statistics

1.1 APPROXIMATION FOR INVERSE HESSIAN

von Neumann Approximation When we apply the von Neumann series for inverse operators on the matrix $I - H$ we get:

$$H^{-1} = (I - (I - H))^{-1} = \sum_{i=0}^{\infty} (I - H)^i \approx \sum_{i=0}^{K} (I - H)^i$$

This is convergent for matrices H with singular values less than 2. An approximation is obtained by truncating the series. While it is invalid for general matrices, this approximation has been shown useful when used in the context of gradient based methods [Lorraine et al., 2020]. To do so one preconditions the matrix H with a suitably chosen large divisor.

Note that the first order approximation is linear in H and along with automatic differentiation methods allows easy and efficient multiplication with any vector by the Hessian-vector product (HVP) method [Christianson, 1992]. To compute the Hessian-vector product (HVP) with the vector v, one simply changes the parameters by ϵv (for some small ϵ) and computes the gradient. The difference between the two gradient when scaled equals the HVP. Furthermore this also holds when multiplying with the cross-Hessian $\partial_\theta \partial_\phi$, the same trick can be used once again. Next the terms in the von-Neumann series can be iteratively obtained by using HVP with the output of the previous iteration. This allows us to compute the series approximation to as many orders as desired. For further details refer to Christianson [1992]

2 DATASET DETAILS

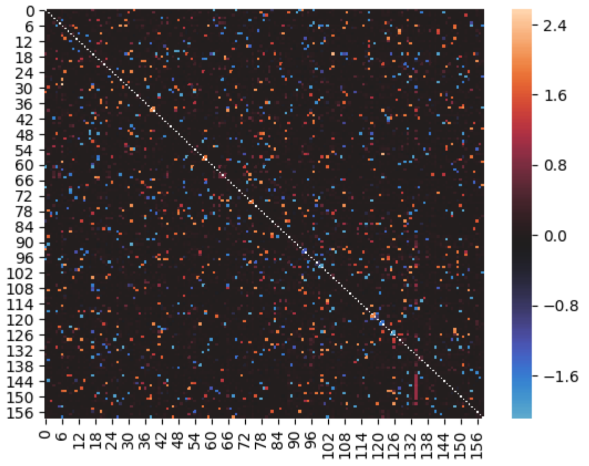
For the larget textual datsets, we follow the processing of [Yang et al., 2018] to preprocess the datasets. We filtered the dataset to 50000 words, and any texts longer than 500 words were discarded. For the smaller MLC datasets we used the standard splits. The details for both are presented in Table 3.

For (POS) tagging, we follow [Tu et al., 2020] and use annotated datset from [Owoputi et al., 2013]. The data set has 25 output tags. We also conduct experiments with small scale image segmentation on the Weizmann horses dataset [Borenstein and Ullman, 2002]. This is a classic dataset for structured prediction evaluation. It contains 328 images of horses and their manually labelled segmentation masks. For this task we follow the protocol detailed in Lu and Huang [2020].

3 ANALYSIS OF LEARNT ENERGIES

3.1 MLP CLASSIFICATION

The role of the global energy function $v^T\sigma(My)$ in E_ϕ is to model interaction between labels. The gradient of E wrt the label y has contributions from values of other labels, and optimization of E should correspondingly increase or decrease the likelihood of a label, given the current probabilities of other labels. To test this hypothesis we compare the Hessian of the learned global energy wrt the output y. For frequently co-occurring labels y_i, y_j, increasing y_i should give positively impact the gradient of E wrt y_j. Similarly for pairs which co-occur less frequently the hessian should give negative values. In Figures 1 we plot the average Hessian of the energy function over the instances as well as the co-occurence matrix of the labels. Note that the diagonals have been removed. We see a general correspondence between the co-occurence matrix and the hessian, though there are values which do not correspond.


(a) Energy Hessian

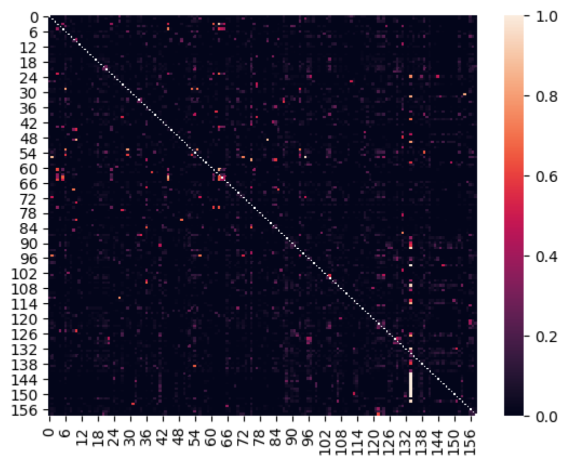

(b) Co-occurence Matrix

Figure 1: Learnt energy and Label Cooccurence on bibtex

3.2 POS TAGGING

In Figure 2 we plot average cross-Hessian of the energy function E wrt y_t, y_{t+1} in the POS tagging experiment. That is we are plotting $H = \frac{\partial^2 E}{\partial y_t \partial y_{t+1}}$. Ideally the model should learn to put weight on tag pairs which follow each other i.e. if the tag A frequently appears after tag B, the term in H_{BA} should be higher. Our experiments suggests this to be the case. As an example the tag 0 corresponds to nouns, 5 to verbs and 6 corresponds to adjectives. We can see from the map that the matrix downweights a noun-adjective and adjective-verb pairing; while upweighing the adjective-noun and noun-verb pairs.

4 ADDITIONAL DETAILS

Addition of task-loss in contrastive divergence The standard constrastive divergence [Hyvärinen and Dayan, 2005] depends only on the energy of positive and negative samples. In this case however we have added the task-loss s to the negative samples, as it more strongly penalizes the energy of model outputs with high task loss. If we consider only one negative sample, then $\mathcal{L}_{CD}$ reduces to:

$$\log \frac{1}{1+\exp(E_\phi(x,y) - E_\phi(x,\bar{y}_1) + s(\bar{y}_k, y)))} = \text{softplus}(s(\bar{y}_k, y) + E_\phi(x,y) - E_\phi(x,\bar{y}_1))$$

which is essentially the same as the margin loss $\mathcal{L}_{SSVM}$ with the hinge being replaced by softer function. Furthermore similar to NCE[Gutmann and Hyvärinen, 2010] and CD[Hyvärinen and Dayan, 2005] losses, by using multiple samples it can provide greater information for the energy function, which is not possible for the SSVM loss.

5 ADDITIONAL EXPERIMENTS

Following Tu and Gimpel [2019] we use our proposed method for a POS-tagging task as well. We use the Twitter-POS data from Owoputi et al. [2013]. The energy model E_ϕ is similar to the one used for NER tasks. We compare against a Bi-LSTM and CRF baseline, SPEN [Belanger et al., 2017] and InfNet [Tu and Gimpel, 2019] based model. Our results are presented in Table 4.

We also conduct experiments with binary image segmentation on the Weizmann horses dataset. Following Pan et al. [2020] and we resize the images and masks to be 32 × 32 pixels. For the inference network we used a convolutional model with the same design as in cGLOW [Lu and Huang, 2020] like . As baselines we compare with DVN [Gygli et al., 2017], ALEN [Pan et al., 2020], and cGLOW [Lu and Huang, 2020]. The task loss in this case the the IoU (intersection over union) metric. Our results are reported in Table 5. It is clear that our proposed method achieves the highest IoU among the comparison methods with a close to 4% percent improvement over ALEN, which itself is much ahead of other models.

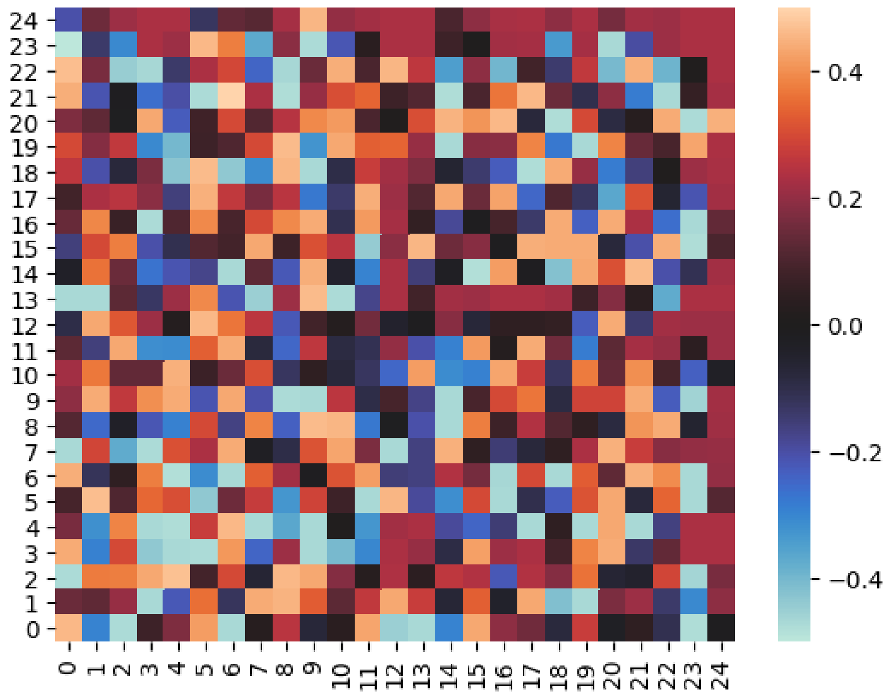


Figure 2: Heatmap representation of learnt pair correlation energies for POS tagging

Models	Accuracy
BILSTM	88.7
SPEN	88.6
CRF	89.3
InfNet	89.7
$\mathcal{L}_{\text{NCE}}$	**90.1** †

Table 4: Test results for POS Tagging with different energy based models. † indicates statistically significant

Model	Mean IoU (%)
DVN	83.9
cGLOW	81.2
ALEN	85.7
$\mathcal{L}_{\text{NCE}}$	**89.4** †

Table 5: Mean IoU results for segmentation with different models on the Weizmann Horses dataset. The input image size is 32x32 pixels. † indicates statistically significant