Notes for 12/21/18 Meeting

Adversarial Training Methods for Semi-Supervised Text Classification

Focus of paper: Evaluating usage of adversarial/virtual adversarial modification to embedding output layer of text classifier, as a way to improve robustness and stabilize classification function. This is an extension to the standard usage of adversarial input for images, and is necessary because of the discrete nature of text.

Adversarial training requires labelled data, as it adds a term to the loss function based on the change in probability of the true label from the adversarial perturbation (selected to maximize the change). The loss function then tries to be minimally changed from adversarial perturbations.

Obviously, when used with images the adv. pertub. is added directly to the input data, but in the case of text with discrete input, this is impossible. Instead, it is added to the output of the embedding layer.

This was the first work to apply AT/VAT to training a text/RNN model, and the addition of the single hyperparameter epsilon (the size of the adv. perturbs.) was sufficient to allow state-of-the-art classification.

Model they used was standard RNN/Bi-directional LSTM, trained at sentiment classification on IDMB, Elec, and Rotten Tomatoes. Trained on DBpedia and RCV1 for topic classification.

Details on adversarial: add -log p(y | x + r_adv) from cost where r_adv is picked to make the probability of p(y|x + r_adv) as large in absolute value as possible, conditional on the norm of r_adv being less than epsilon. Obviously, identifying the exact r_adv vector is intractable, so we instead linearize the conditional log probability around x by taking the gradient and choosing r_adv as the vector of size epsilon opposite the gradient.

For virtual adversarial training, it is no longer necessary to use the label. Instead of seeking to minimize the change in classification for the correct class, the loss functions adjoined with the KL divergence between the class probability distribution conditioned on the unaltered input, and the input altered by an adversarial perturbation vector chosen to maximize the KL div. Requiring only the input, VAT can be used for semi-supervised learning.

For AT and VAT, as noise scale is fixed by epsilon, also need to normalize embedding vectors for that to be meaningful.

Also compare to adding noise perturbations - Adversarial and VAT greatly outperform those.

For Elec and RCV1, VAT got best performance, over other more complicated state of the art models.

VAT improved both classification performance and quality of word embeddings. VAT was also best performing in the Realistic evaluation of Semi-Supervised Learning Paper

Takeaways: VAT can be extended to text-based problems, by adding noise post-embedding. It continues to work very well, and requires only an adjusted loss function and one hyperparameter. It can be interpreted as robust optimization and works on SSL problems. Could potentially be combined with

GANs to further improve off manifold classification performance during generation. Or perhaps some "post-training" when the classifier is trained on the supervised + augmented data, adding in the VAT loss. During generator train time, probably don't want to use VAT as classifier performance is part of generator loss.

Deceptive Opinion Spam Detection Using Neural Network

Focus of paper: Using deep NNs for opinion spam identification, showing it leads to improved performance over shallow feature-based ML. Combining the features works even better.

Opinion spam is difficult even for humans to detect, one test had average accuracy of 57%. The point of the paper is that opinion spam is often driven by global semantic features, which NNs are better equipped to pick up over hand-crafted ones.

Utilization of three levels of representation - Word embedding, sentence embedding, doc embedding.

They use 3 stages: first a CNN is used to produce sentence representations from concatenated word vectors. Next, those sentence vectors are fed through a bi-directional RNN with attention, outputting a document vector. Finally, that document vector is optionally combined with manually engineer document features and fed through a dense layer into a 2-class softmax.

Utilize 3 datasets (Li et al 2014) Hotel, Restaurant, Doctor. The Hotel dataset is the same as the Deceptive Opinion Spam Corpus, which is readily available and the largest of the three. It consists of 1600 reviews, 800 real from TripAdvisor and Yelp and 800 fake ones created by Amazon Turkers.

[Note, check out Kim et al 2015 Frame-based semantic feature based on FrameNet]

Although they used pre-training to set up the sentence embeddings, did not use SSL on unlabeled reviews for training class accuracy. The used softmax attention at the sentence vector level and found that it was very important for accuracy on this task.

Standard AdaGrad training and cross-entropy loss with l2 regularization. Ultimately, they found that their deep neural network outperformed by a fair margin previous classifiers trained on hand-made features across all of their opinion spam datasets. Adding the hand-made features as additional input to the final layer provided a modest boost in accuracy as well.

Toward Controlled Generation of Text

Core idea: Use of variational autoencoders and separate discriminators (and a complicated gradient system) to enforce seperable/independant latent representation of constructs of interest, allowing for fine-tuned control over the generated sentences.

Goal: generate realistic sentences whose attributes can be manipulated through disentagled latent representations.

Have to content with discrete visible problem: use of policy learning or continous approximations have high variance, use of element wise reconstruction loses wholistic continuity.

Their solution to discrete problem is softmax approximation produced by generator which anneals to discrete case. This is possible because of the VAE structure, which uses reproduction accuracy as the metric to train on, instead of trying to fool the discriminator. Has fast convergence.

Generator and discriminators effectively mutually bootstrap each other (extension of wake-sleep procedure). Requires very little annotated data, and the data CAN BE DISTINCT for each of the desired controllable attributes.

Distinct from standard "reconstruction" methods, because can use separate labelled data. Discriminators are learning signals, not just reconstruction. Discriminators also allow for SSL.

Their example was generating sentences with simultaneously controlled sentiment and tense.

Wake sleep algorithm, similar to VAEs by combining inference network with generator: wake phase updates generator with samples generated based on discriminator network on train data,sleep phase updates discriminator network with samples from generator. Extended wake sleep updates both generator and discriminators

Disentagled latent representations is powerful, see InfoGAN in image space

Controllability comes from concatenating latent vector from encoder with attribute coding, and enforcing independence constraints via reproduction.

x (input sentenct) -> encoder -> z (latent) + c (attr coding) -> generator -> x_hat

(generated sentence, actually approximate softmax output).

Typically VAEs are trained by RMSE (l2 norm) in data space between input and generator output, trying to mininimize the reconstruction error. Instead, computing distances in feature space (that is in the latent post-encoding vector) is seen to be robust to transformations and is better.

That is, discriminators take generated soft sentence, and try to predict the sentence code for each attribute (diff discriminator per attr). Error can be backpropagated directly into generator.

To enforce independance of latent z and coding c, use encoder as additional discriminator, taking the soft generated sentence x_hat and trying to predict z without c. Thus, varying attribute codes should keep unstructured attributes invariant as long as c is not changed.

VAE here has standard reconstruction training, as well as discriminators which are additioanlly trained on real corpuses with known coding c. The combined VAE/wake-sleep algo is very efficient SSL training for discriminators.

In detail:

Generator is LSTM-RNN generating tokens conditional on z, c, and previous x_hat tokens.

z is modelled as normal, CTS variables with gaussian prior. C can have both continous and disrete attributes with appropriate prior. Encoder creates probabilistic coding z conditional on x. VAE has L_vae loss component as is standard, with loss minimizing the reconstruction error probability on real sentence while also remaining close to prior via KL divergence.

Thus, reconstruction loss pushes generator to produce realistic samples, discriminators provide extra learning signal for attribute production.

L_attr_c for generator causes improved performance of generator at generating sentences which the discriminator classes as being of the desired class c.

L_attr_z is maximizing the likelihood of producing z conditional on the generated sentence, thus ensuring that the latent variable z is independent of the attr coding.

Combined, the loss function of the generator is minimizing L_G = L_VAEreconstruction + L_attr_c + L_attr_z with balancing coefficients.

Encoder is trained solely by minimizing L_VAE reconstruction

For learning each discriminator, we need a set of labelled examples x_l and c_l Then the discriminator has L_s which aims to maximize the probability of the correct label conditioned on the true sentence.

Additionally, the generator also produces noisy sentence attribute pairs x_hat, and c. We can use that, when a entropy regularization term is added, to help train the discriminat or. L_u which takes the probabiliy of producing the correct clas balanced with the shannon entropy, which pushes the discriminator to have high confidence in label preds. The discriminator is then trained on the loss L_d = L_s + L_u with balancing params.

The training procedure is as follows: Base VAE is trained on large corpus of unlabelled sentences, minimizing reconstruction expected error (c is sampled from prior for this, playing no part).

The full model is then trained by alternating sleep and wake phases. In the sleep phase, samoes are produced from the generator and used as noisy targets for the discriminators as in L_u. Still in the sleep phase, we can use the generated sentences to improve the generator itself (ancestrally) via passing the generated samples through L_attr,c and L_attr,z.

The wake phase in then uses the VAE error, sampling c conditioned on the true sentence and passing it through the encoder and generator, optimizing L_VAE.

They train this on some sentiment and tense labeled datasets, as well as the large unlabeled IMDB corpus. They find that they are able to generate quite realistic sentences and have a fine-grained control over the tense and sentiment of the output.

[Hu 2017 unified representation of deep generative networks]