Sensorveiledning TEK5040 H2019

1 Semantic segmentation (weight 7%)

Assume that you are training a segmentation model to segment out the lines that mark the different lanes on a road. Everything except theses lines are treated as the background class. Assume that you obtain 99 % accuracy on the test set. What is an undesirable property of accuracy as a measure of performance in this example? Give at least two examples of alternative metrics that can be used.

Solution: Very large class imbalance. One may probably get this accuracy by just predicting that there is not a line. *Precision* and *recall* are good alternative metrics to use.

Comments:

- 4 points to note that class imbalance makes it possible to get good score by only predicting background.
- 1.5 points each for precision, recall, intersection over union and other reasonable scores. Maximum of 3 points.

2 Image captioning with attention (weight 12%)

Assume that you want to train a Recurrent Neural Network (RNN) to generate captions for images, and that you have a pretrained CNN that extracts feature maps of dimension $h \times w \times c$ from the images, where h, w and c are the height, width and number of channels of the feature maps. Describe one way you could use content-based attention to focus on a particular part of the image at each time step.

Solution: Example with multiplicative attention: At each time step t, generate a query vector q_t from state s_t, e.g. by applying a matrix Q, such that $q_t = Qs_t$. For each of the $h * w$ locations in the image, apply a key matrix K such that $k_i = Kx_i$ where i is a location and x_i is the c -dimensional vector of features at that location. You could then take the inner product between each of the key vectors and the query vector, and proceed by normalizing with the softmax function to get a probability distribution over the locations. Using soft attention, you could then get the desired vector by a weighted mean over the features at the the different locations in the image, where the weight is given by the probability p_i calculated for that location.

Comments: 12 points for an acceptable example (looked at dot-product (multiplicative) and Bahdanau (additive) in exercises). 6 points if e.g. describing how this fits into the framework of content-based attention without giving an example of where the 'keys' and 'queries' comes from.

3 Greedy policy update (weight 7%)

Assume that we have a policy π and let v_π and q_π denote the state-value and action-value function respectively. Assume a discrete action space. Define a new policy π' by

$$\pi'(s) := \text{argmax}_a q_\pi(s, a)$$

i.e. greedily choose the action that looks the most promising. Is it possible that the new policy is *worse* for some states, i.e. that $v_{\pi'}(s) < v_\pi(s)$ for some state s? Justify your answer.

Solution: No, the *policy improvement* theorem states that this is impossible.

Comments: 0 points for wrong answer, or correct answer but no or wrong justification of answer.

4 Proximal Policy Optimization (weight 10%)

Let π_θ denote the policy parametrized by $\theta \in \Theta$. Assume that we have sampled a number of episodes using the weights θ_{old} for our policy. Define

$$u_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \tag{1}$$

and let $\hat{d}_t$ denote the estimated advantage of taking action a_t from state s_t. The PPO surrogate objective is then defined as

$$L^{PPO}(\theta) = \hat{E}_t[\min\left(u_t(\theta)\hat{d}_t, \text{clip}(u_t(\theta), 1-\epsilon, 1+\epsilon)\hat{d}_t\right)] \tag{2}$$

where $\epsilon > 0$ is a hyperparameter and $\hat{E}_t$ denotes the expectation over the sampled dataset.

What is the purpose of the hyperparameter ϵ used in clip operation?

Solution: The unclipped objective of PPO is an estimate of the expected change in returns by changing the policy from $\pi_{\theta_{\text{old}}}$ to π_θ. This approximation however ignores the change in discounted state-visitation frequencies and can not be trusted if we change our policy too much. Clipping the probability ratio removes the incentive for moving u_t outside of the interval $[1\epsilon, 1+\epsilon]$, i.e. changing the policy too much

Comments:

- 7 points if mentions that it is to ensure that we don't change policy too much.
- 2 points if also mentions that it does so by moving the *incentive* to change policy too much (and not actually enforcing it).
- 1 points to mention why this is desirable, i.e. approximation state-visitation frequencies.

5 Meta learning (weight 12%)

Tom plans to train an image classifier with a training set which contains many images of 100 classes. He also has a test set consisting of images of all the 100 classes. Bob gets this classifier and tries to use it to classify a given image into one of the three new classes {*truck, train, motorcycle*}. Unfortunately, Bob has only one example of each class. Bob and Tom decide to use meta learning with the Matching network architecture to tackle the problem.

a. (5%) If the task of Tom was a regular training, a data sample would be $(\boldsymbol{I}, \boldsymbol{c})$, where $\boldsymbol{I}$ is a sample image and $\boldsymbol{c}$ is its class label. In meta learning, what is the structure of a training data sample Tom can use? (Hint: Think of a meta-training set)

Solution: Each data sample of the meta-training set consists of a pair of a support set and a query set. A support set consists of $(\boldsymbol{I}_1, \boldsymbol{c}_1, \boldsymbol{I}_2, \boldsymbol{c}_2, \boldsymbol{I}_3, \boldsymbol{c}_3)$, where $(\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3)$ are three distinct sample class labels drawn from the training class labels and $(\boldsymbol{I}_1, \boldsymbol{I}_2, \boldsymbol{I}_3)$ are training image examples of classes $\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3$ respectively. The query set consists of single image example and its class label drawn from the test set.

Comments:

- 2 points for mentioning support and query set (equivalent names are accepted).
- 2 points for correct support set. Fraction of points can be given depending on the degree of correctness.
- 1 point for the correct query set.

b. (4%) What is the structure of a data sample Bob can use? (Hint: Think of a meta-test set)

Solution: Each data sample of the meta-test set consists of a support set and a query set. The support set consists of three image examples with known, distinct class labels drawn from the set {*truck, train, motorcycle*}. The query set consists of an image example of unknown class label drawn also from the set {*truck, train, motorcycle*}.

Comments:

- 1 point for mentioning support and query set (equivalent names are accepted).
- 2 points for correct support set. Fraction of points can be given depending on the degree of correctness.
- 1 point for the correct query set.

c. (3%%) The Matching network outputs a probability vector $[p_1, p_2, p_3]$. What do p_1, p_2 and p_3 represent?

Solution: p_i represents the similarity (measured through the cosine distance) between the query image and the i^{th} class of the support set.

Comments:

- Full marks for any answer indicating that p_1, p_2 and p_3 represent the similarity between the query image and the respective support images.
- No marks for just mentioning probabilities.

6 Processing 3D-Scenes (weight 12%)

a. (2%) One approach for segmentation and classification on 3D scenes is to apply 3D-convolutions on voxelized point clouds. Briefly describe a problem with this approach.

Solution: Voxelization tries to divide the volume into equal sub-volumes irrespective of whether there is a point in it or not. This is a highly inefficient representation because voxels can be allocated to empty space. This leads to high memory requirements and hence achievable resolution of representation can be lower.

Comments:

- 2 points for mention of any problem: inefficient,too much empty space, high memory requirements or lower resolution possible.

b. (4%) Give a short explanation of how the Pointnet achieves permutation invariance.

Solution: Point net processes all points independently and finally applies a symmetric function (max-pooling)

Comments:

- 1 point for independent processing.
- 3 points for either symmetric function or max-pooling

c. (6%) Using graph convolutions, we can exploit the information of local structure in point clouds. A general graph convolution called *edge convolution* can be defined using the function $\boldsymbol{h}_\theta(\cdot,\cdot)$ which is parameterized by θ:

$$\boldsymbol{x}_i' = \underset{j\,:\,(i,j)\,\in\,\mathcal{E}}{\square}\ \boldsymbol{h}_\theta(\boldsymbol{x}_i, \boldsymbol{x_j}),$$

where $\square$ denotes the aggregation operation for the node i over all its neighbours j, whereas $\boldsymbol{x}$ and $\mathcal{E}$ denote feature vectors and the set of all edges in the graph respectively. Show that the regular convolution used in CNNs and operations used in the Pointnet are special cases of this edge convolution.

Solution: When the aggregation operation is summation and the edge function $\boldsymbol{h_\theta}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{\theta} \cdot \boldsymbol{x}_j$, the edge convolution becomes regular convolution. When there is no aggregation and $\boldsymbol{h_\theta}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{h_\theta}(\boldsymbol{x}_i)$, it becomes the operations in Pointnet.

Comments:

- 3 points each for correct explanation of regular convolution and Pointnet operations. Either mathematical or verbal answers are accepted.
- Regular convolution
  - 1 point for aggregation = summation
  - 2 points for edge function is the dot product.
- Pointnet operations
  - 1 point for no aggregation
  - 2 points for edge function is only a function of the node concerned (i.e. neighbour nodes are not considered).

7 Tracking in video (weight 11%)

a. (2%) Name four basic operations in multiple object visual tracking (MOT) in which deep learning can be employed.

Solution: Detection, feature extraction, affinity computation, association or track management

Comments:

- 0.5 points for each operation.

b. (4%) Why can a siamese network be a simpler and more efficient way of tracking objects in a video compared to MDNet?

Solution:

- You only need to run the entire network *once* for each input image.
- No runtime training

Comments:

- 2 points for running the network once
- 2 points for no runtime training

c. (5%) A problem with siamese networks is that they do not handle different scales of the tracked object well. Give an outline of a method with which we can improve on that.

Solution: One can use a region proposal network to calculate the bounding boxes through regression and the probability of match/no match through binary classification. Predicted bounding boxes can represent different scales.

Comments:

- 4 points for mention of region proposal net and 1 point for mentioning that it predicts bounding boxes of different scales
- Alternative methods such as using several scales in image matching is also acceptable.

8 Conditional GAN (weight 10%)

Assume that you have a set of color images $x_1, x_2, \ldots, x_N$ and you want to train a neural network to *colorize* grayscale images, i.e. so that given a grayscale image x' the network may generate different plausible color images. As an application, you could then use this network to possibly colorize old digitized grayscale photos. Briefly describe how you can formulate this as a conditional GAN problem.

Solution: For each color image x we can construct a grayscale image x' by taking a weighted mean over the color channels. We then get a paired dataset $(x_1, x'_1), (x_2, x'_2), \ldots, (x_N, x'_N)$, where the grayscale image, x'_i, is the condition variable.

Comments:

- 5 points to identify the grascale image as the condition variable.
- 5 points to propose how we can get paired samples for training.

9 Evidence Lower Bound (weight 7%)

a. (3%) In variational inference we try to approximate the posterior distribution $p(\boldsymbol{w}|\mathcal{D})$ with an auxiliary distribution $q(\boldsymbol{w})$, where $\boldsymbol{w}$ and $\mathcal{D}$ are network parameters and training data respectively. The KL divergence between the two distributions is given by

$$\mathrm{KL}\big(q(\boldsymbol{w})||p(\boldsymbol{w}|\mathcal{D})\big) = -\mathbb{E}_{q(\boldsymbol{w})} \ln \frac{p\left(\boldsymbol{w}, \mathcal{D}\right)}{q\left(\boldsymbol{w}\right)} + \ln p(\mathcal{D})$$

Identify the quantity known as Evidence Lower Bound (ELBO) in the above equation.

Solution: ELBO $= \mathrm{E}_{q(\boldsymbol{w})} \ln \dfrac{p\left(\boldsymbol{w}, \mathcal{D}\right)}{q\left(\boldsymbol{w}\right)}$

Comments:

- 3 points for correct identification.

b. (4%) Explain why maximizing ELBO would lead the distribution $q(\boldsymbol{w})$ to get closer to $p(\boldsymbol{w}|\mathcal{D})$.

Solution: Since $\ln p(\mathcal{D})$ does not depend on $q(\boldsymbol{w})$ or $p(\boldsymbol{w}|\mathcal{D})$, maximizing ELBO would lead to minimization of KL-divergence between $q(\boldsymbol{w})$ and $p(\boldsymbol{w}|\mathcal{D})$. That means that $q(\boldsymbol{w})$ and $p(\boldsymbol{w}|\mathcal{D})$ get closer. variable.

Comments:

- 2 points for identifying that $\ln p(\mathcal{D})$ is constant.
- 2 points for mentioning that KL is minimized.

10 Deep learning for control (weight 12%)

a. (2%) What is the main motivation for inverse reinforcement learning?

Solution: In complex real world control situations, it is difficult to design good reward functions. Learning reward function is an alternative and hence inverse reinforcement learning.

Comments:

- 2 points for the mention of difficulty in designing good reward functions for some real world problems.

b. (5%) Assume you are given a set of expert demonstrations, a policy network and a reward network. Both network are initialized to random values. The Guided Cost Learning (GCL) algorithm is used to train the policy and reward networks. Write down a very high level of pseudo code of the GCL algorithm. (Hint: Show only the three basic steps).

Solution:

1. Run the policy on the environment. Generate the sample trajectories.
2. Optimize the reward function with sample trajectories and expert demonstrations as inputs and the GCL loss as the objective function
3. Use the optimized reward function and run a reinforcement learning step to optimize the policy network. Go to step 1.

Comments:

- 1 point each for mentioning basic steps: run policy, optimize reward function and optimize policy function.
- 2 points for mentioning that GCL loss is used in reward optimization
- Any description containing these elements are acceptable.

c. (5%) Modify your pseudo code to that of Generative Adversarial Imitation Learning (GAIL).

Solution:

1. Run the policy on the environment. Generate the sample trajectories.
2. Optimize the discriminator function to minimize the probability of sample trajectories and maximize the probability of expert demonstrations.
3. Use the discriminator output as the reward and run a reinforcement learning step to optimize the policy network. Go to step 1.

Comments:

- 1 point each for mentioning basic steps: run policy, optimize discriminator function and optimize policy function.
- 2 points for mentioning that discriminator loss is to distinguish expert demonstrations against the sample trajectories.
- Any description containing these elements are acceptable.