CS 7150: Deep Learning — Summer-Full 2020 — Paul Hand

Week 12 — Preparation Questions For Class
Due: Monday July 27, 2020 at 12:00 PM Eastern time via Gradescope

Name: Saurabh Vaidya
Collaborators: [Put Your Collaborators Here]

You may consult any and all resources in answering the questions. Your goal is to have answers that are ready to be shared with the class (or on a hypothetical job interview) as written. **Make sure to tag each question when you submit to Gradescope.**

Directions: Read the article 'Density estimation using Real NVP'.

Question 1. *Compare and contrast RealNVP with VAEs and GANs.*

Response:

Answer 1. *Due to change of variable formula and normalizing flows, RealNVP can allow for exact log likelihood of data. In VAE, we approximate this log likelihood by maximing a lower bound on the likelihood and in GANs we altogether avoid computing maximum likelihood and instead optimize the minimax criterion with a discriminator and generator.*

VAE and GANs both learn the data distribution through a latent space of lower dimension whereas in RealNVP the latent space has the same dimensionality as data. Unlike VAEs, RealNVP does not depend on reconstruction error costs. And unlike VAEs and GANs, realnvp is able to learn semantically meaningful latent space of the same dimensions as input space.

Question 2. *Explain how the RealNVP is invertible, even though it depends on convolutional neural networks, which in general are not invertible.*

Response:

Answer 2. *RealNVP makes use of change of variable formula which use bijective functions and they are invertible. Change of formula involves computing Jacobians of bijective functions. However, computing Jacobians of functions with high dimensional domains is expensive. Thus easy computation of jacobians is crucial to invertibility.*

To solve this, authors make use of coupling layers where they split input vector into two halves. The first half is applied a direct mapping while the remainder is applied a complex transformation which is still bijective.

This complex transformation usually is a scale(multiply) and translation(add) operation which can be inversed by negative scaling(division) and reverse translation(subtract).

$$y_{1:d} = x_{1:d} \tag{1}$$

$$y_{d+1:D} = x_{d+1:D} \odot \exp(s(x_{1:d})) + t(x_{1:d}) \tag{2}$$

where s, t are CNN.

As can be seen, the Jacobian of coupling layer does not involve computing Jacobian of s, t. Moreover, inverting the coupling layers also does not involve inverting s or t as seen in $x_{d+1:D} = (y_{d+1:D} -) t(y_{1:d}) \odot \exp(s(y_{1:d}))$ where the input to s and t is $y_{1:d} = x_{1:d}$. As the input is same during forward propagation and inverse the output of CNNs remains same and thus we do not have to invert them.

Thus even though RealNVP depends on CNNs, due to design of coupling layers we never have to invert CNNs.

Question 3. *Explain the multiscape architecture of the RealNVP.*

Response:

Answer 3. *A multiscale architecture of RealNVP consists of multiple layers stacked together. A single scale(layer) consists of sequence of 3 operations: 3 coupling layers with alternating checkerboard masks, squeezing of sxsxc tensor into $\frac{s}{2}$x$\frac{s}{2}$x4c tensor and another 3 coupling layers with alternating channel wise masking. These scales(layers) are stacked to form a multi scale architecture.*

In each layer though half of the D dimensions of input are factored out as output and remaining half are pushed to next layer. This saves up on memory and computations. Thus the final output is all these variables factored out at different layers concatenated together.

The outputs from each layer are Gaussianized in sequence of earlier layers first to later layers. As a result we get different levels of representation of latent space at each layer giving more fine-grained features.

Question 4. *Is training an invertible network like RealNVP a supervised or an unsupervised machine learning problem?*

Response:

Answer 4. *Training invertible networks like RealNVP involve training procedure to learn a latent space such that inverse of that latent space would maximize the data likelihood. As like any generative model we are only given observed data and our goal is to learn a the data distribution. As this does not involve any target variables / learning a target posterior distribution, this training regime is unsupervised. We are given a data and without any supervision we learn the distribution of the data by maximizing likelihood.*

Question 5. *How does the RealNVP ensure that any input component can affect every output component?*

Response:

Answer 5. *RealNVP makes use of alternating checkerboard masks on input along with alternating channels masks as shuffling components to divide the input vector in two parts. Since we divide our inputs in 2 parts and each part goes through 3 coupling layers followed by shuffling and 3 more coupling layers, the corresponding 2 output components are combination of all input components. This sequencing of a input component through coupling with alternate masking increases the receptive field of the output to include more input components.*

Thus every input goes through identity and scaling+translation transformations thereby affecting every output component/