CS4851/6851 - Intro to Deep Learning

Homework - 5

Student Details:

Level-of-Study	Name	Panther-ID	Email ID
Graduate Student	Kiran Kumar Reddy Donuru	002678089	kdonuru1@student.gsu.edu

1)For each of the following norms, explain what properties will they favor when used in reconstruction error: L0, L1, and L2.

Firstly to answer this question. I will need to understand what exactly is reconstruction error.

Reconstruction error: Reconstruction error is the deviation between test data and the corresponding reconstructions computed by a model trained on a reference dataset.

Now starting with each of the norms L0, L1, and L2 will try to explain what properties will they favor.

L0 norm properties that favor when used in reconstruction error:
All non-zero weights are counted in the L0 norm. The amount of non-zero parameters might be a useful selection characteristic when using this method. This characteristic is useful for identifying just the countable values of the reconstruction error that differ. The zero values would be ignored in this case, and only the non-zero values would be prioritized. This can also aid parameter selection, such as disregarding factors that have little or minor influence on the result and picking only the decision-making characteristics.

L1 norm properties that favor when used in reconstruction error:
The absolute total of all weights is used as the L1 norm. This feature appears to have equal weighting for all mistakes, whether they are outliers or not, at first look in terms of the reconstruction error. The parameters' size and value are proportional to the model's complexity. As a result, complicated models have a large L1 norm, which leads to a large loss function, suggesting that the model is insufficient. This may be useful if we don't want to place as much focus on outliers in our reconstruction error estimations. L1 can also generate a variety of solutions. L1 may not avoid overfitting, but it will give essential characteristics greater weight, and parameters will be normalized according to their scale. Its assistance is used to pick features. To conclude, because of its sparsity, the L1 norm may be more advantageous.

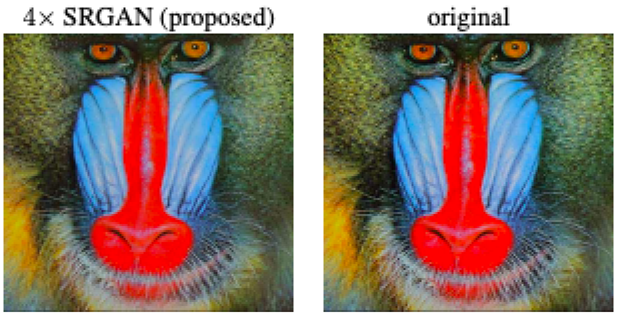
L2 norm properties that favor when used in reconstruction error:
When the weights are near to zero and below one, L2 will reward the model. As a result, their squares will be almost nil. Unlike the L1 norm, the L2 norm employs non-zero coefficient regularization methods such as ridge regression. It's the total of all the weights in the model's squares. In comparison to L1, L2 norms punish outliers due to squaring. This is useful if we are very sensitive to outliers and want to see them appear more frequently in the reconstruction error. The attribute of L2 tries to reduce loss, which is a good thing. While L2 is more computationally challenging than L1, new technology alleviates these issues, resulting in greater predictive characteristics than L1. L2 will keep the model from overfitting.

2)Given a set of contrast images with sharp geometric edges (e.g. photo-lithography masks for microprocessor manufacturing) write down a formulation for reconstruction error that would work best. Justify your choice.

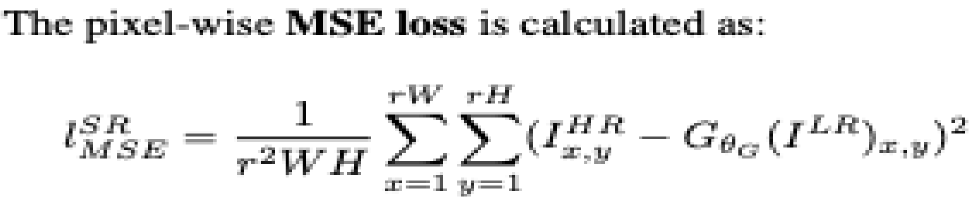
High contrast geometric images are highly complicated, requiring penalties from the reconstruction error for big enough distances in the pixel-to-pixel comparison. This characteristic will result in a simple solution. To develop a loss function for a high-resolution color image with a natural edge, numerous elements must be considered. Because it is the most punishing, the L0 norm would be best. Using the L0 norm, which has the feature of simply distinguishing non-zero elements, aids in swiftly determining whether or not a pixel has been wrongly rebuilt.

3)Given a set of images of wildlife taken in their natural habitat write down a formulation for reconstruction error that would work best. Justify your choice.

Solution-Source:
https://openaccess.thecvf.com/content_cvpr_2017/papers/Ledig_Photo-Realistic_Single_Image_CVPR_2017_paper.pdf


4× SRGAN (proposed)
original

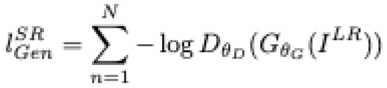
Content loss: A loss function that is closer to perceptual similarity than pixel-wise losses can be used. It is estimated the updated Euclidean distance between the feature and the GAN loss.


The pixel-wise **MSE loss** is calculated as:

$$l_{MSE}^{SR} = \frac{1}{r^2WH}\sum_{x=1}^{rW}\sum_{y=1}^{rH}(I_{x,y}^{HR} - G_{\theta_G}(I^{LR})_{x,y})^2$$

W and H are the feature maps of each node, respectively. This is a total loss divided by the weight and output of each node. This helps to maintain the loss dispersed throughout the whole picture while also allowing each node to learn. Rather than having to compute the entire picture loss and then learning from the NN.

Adversarial loss: In addition to the content loss, we add the generating component of our GAN to the perceptual loss. Trying to fool the discriminator network leads to our network favoring solutions based on natural imagery.


$$l_{Gen}^{SR} = \sum_{n=1}^{N} -\log D_{\theta_D}(G_{\theta_G}(I^{LR}))$$

We utilize Adversarial loss since GAN cannot be trained just on content loss, and we don't anticipate the GAN to produce images from our natural photos. This loss aids the model's learning by allowing it to construct images that fit into the pool of natural images, fooling the discriminator network. The likelihood of the created picture D(I) appearing in the pool of natural images is given by D(). As a result, the sum of negative logs has been introduced to optimize the Model.

This method provides some resilience, and it allows for enough relaxation to build HD pictures using GAN, where sharp edges may be found in smaller numbers and address a color image.

Solution with L0,1,2 Norms would be: For example, to calculate reconstruction error in wildlife photographs or any other sort of image where geometrical patterns do not exist, the L2 norm can be used. This is because we can now compute similarities between pixels without having to binary classify them as right or wrong by utilizing the MSE as our error type. When looking at grass pixel by pixel, for example, the space between each grass appears little since they are all close together. One may also argue that L1 is a method of calculating pixel-to-pixel distances with equal weight, as opposed to L2, which penalizes outliers more heavily owing to squaring the disparities.

4)Given distributions p and q. If q is parameterized by θ, how would you choose the value for θ to make q closest to p among all possible q’s?

(a) Write down formulation of how would you measure the closeness of q to p.

An MSE gives the sum of the distance between the points. However, as we can see, a Distributional is a scale and how the points are dispersed to the other points, not a point of measurement. Before we look at the differences, let's look at how to quantify a disturbance.

Entropy: The entropy of a random variable is the average degree of information or uncertainty inherent in the variable's likely outcomes. Assume you have a discrete random variable called x, with p(x) being the distribution's value for x.

```
Entropy  =   sigma(p(x)log(p(x)))
```

Cross-Entropy: Using the idea of entropy from information theory, the amount of bits required to describe or transmit an average event from one distribution compared to another is computed. Cross-entropy may be calculated using the probability of occurrences from P and Q. The problem is that this isn't a comparison of the two Distributional curves, but rather the number of bits required to represent one in place of the other.

```
Cross-Entropy: - sigma( p(x)log(q(x)) )
```

KL Divergence:This metric is similar to cross-entropy in that it measures a quantity. It determines the average number of extra bits required to convey a message using Q instead of P, rather than the total number of bits. The K L divergence is calculated by multiplying the negative sum of the probability of each event in P by the log of the probability of the event in Q over the probability of the event in P.

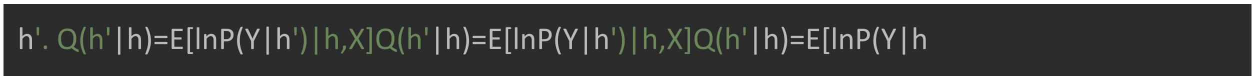
```
KL(p, q) = Sigma( p(x) log (q(x) / p(x)) )
```

The KL divergence is a common information-theoretic "measure" of the difference between two probability mass functions that have been used for distribution functions. The relative entropy, or the difference between cross-entropy and entropy, or any distance between the actual and anticipated probability distributions, is known as KL divergence.

b) Explain what you would do to maximize this closeness (i.e. make q and p maximally close, or minimally different or divergent)

To maximize the proximity, the EM algorithm is utilized. It looks for the h' that maximizes

E[lnP(Y|h')] to get the greatest likelihood hypothesis



h'. Q(h'|h)=E[lnP(Y|h')|h,X]Q(h'|h)=E[lnP(Y|h')|h,X]Q(h'|h)=E[lnP(Y|h

The likelihood of the entire data Y given hypothesis h' is P(Y|h'), thus maximising lnP(Y|h') also maximizes P(Y|h'). Over the probability distribution guiding the random variable Y, we take the expected values E[lnP(Y|h')]. To estimate the distribution governing Y, the EM algorithm employs its current hypothesis h instead of the actual parameters q. The hypothesis h is substituted with h', which maximizes the function h=argmaxh'Q(h'|h). The EM approach converges to a stationary point of the likelihood function P(Y|h') when function Q is continuous.

Reference:
https://towardsdatascience.com/17-types-of-similarity-and-dissimilarity-measures-used-in-data-science-3eb914d2681

5)Write a report on one of the following topics related to GANS.

(a) InfoGAN https://arxiv.org/abs/1606.03657

I have chosen to write a report on the topic InfoGAN.

Introduction: The difficulty of extracting value from vast volumes of unlabeled data is known as unsupervised learning. The goal of representation learning, a popular unsupervised learning technique, is to create a representation that reveals important semantic qualities as easily decodable pieces from unlabeled input. A large percentage of unsupervised learning research is driven by generative modeling. Two of the most prominent generating models are the variational autoencoder (VAE) and the generative adversarial network (GAN). These findings suggest that using generative modeling and a reciprocal information cost to train disentangled representations might be a viable option. Then we'll examine GANs, which are InfoGAN's basis.

Variational Mutual Information Maximization: The mutual information term I(c; G(z, c) is difficult to optimize directly since it requires access to the posterior P (c|x). We can decrease the limit by creating an auxiliary distribution Q(c|x) to approximate P (c|x): Variational Information Maximization is a mutual information limiting method. L I, in particular, has room for improvement. As a consequence, L I (G, Q) may be added to GAN's objectives without changing the training process, and the approach is known as Information Maximizing Generative Adversarial Networks (IMGN) (InfoGAN). When the latent code contains continuous variables, a lower value is employed to guarantee that L I (G, Q), which now incorporates differential entropy, is on the same scale as GAN goals.

Experiments: The main goal of the experiments is to explore if mutual information can be efficiently maximized. The second aim is to test if InfoGAN can acquire disentangled and interpretable representations by modifying only one latent component at a time in the generator to see if changing that factor produces only one sort of semantic variation in produced pictures. This approach is also used by DC-IGN to evaluate their learnt representations on 3D image datasets, which are then compared using InfoGAN. We use the MNIST dataset to train InfoGAN with a uniform categorical distribution on latent codes c Cat(K = 10, p = 0.1) to determine if the proposed technique can efficiently maximize the mutual information between latent codes c and produced pictures G(z, c).

Mutual Information Maximization: We train a normal GAN using an auxiliary distribution Q as a baseline when the generator is not explicitly instructed to maximize mutual information with the latent codes. Because we employ expressive neural networks to parametrize Q, we believe that Q reasonably approximates the real posterior P (c|x). As a result, in normal GAN, there is minimal reciprocal information between latent codes and produced pictures. Even though we have not identified such a situation in our research, we believe that various neural network topologies may contain more reciprocal information between latent codes and output pictures. This comparison demonstrates that in a conventional GAN, there is no guarantee that the generator will use the latent codes.

Disentangled Representation: On MNIST, we used one categorical code, c 1 Cat(K = 10, p = 0.1), to describe discontinuous variation in data, and two serial codes, c 2 and c 3 Unif(1, 1), to model continuous variations. Continuous style modifications are recorded by serial codes c 2 and c 3: c 2 model digit rotation, while c 3 regulates width. To model the latent components in this experiment, we employed four categorical codes (c 1, c 2, c 3, c 4 Cat(K = 20, p = 0.05) and one continuous code (c 5 Unif(1, 1).The Street View House Number (SVHN) dataset was used to test InfoGAN. Because it is noisy, has variable-resolution images and distracting numerals, and lacks numerous variants of the same object, training an interpretable representation is far more challenging. In this experiment, we employ two uniform continuous variables and four ten-dimensional categorical variables as latent coding. We give the representation that most closely follows past supervised findings for each factor out of 5 random runs to enable a direct comparison.

Conclusion: This study introduces a representation learning technique called Information Maximizing Generative Adversarial Networks (InfoGAN). Generator: Instead of drawing from the real data distribution, the "sleep" phase "dreams" up samples from the current generator distribution to update the auxiliary distribution Q. As a consequence, we can see that the "sleep" phase update is the update step in the Wake-Sleep algorithm while maximizing the surrogate loss L I w.r.t. Q. InfoGAN varies from Wake-Sleep when it comes to optimizing concerning LI Because InfoGAN additionally updates the generator during the "sleep" phase, our technique may be thought of as a "Sleep-Sleep" algorithm. This distinction between InfoGAN and other generative modeling approaches is highlighted by this interpretation. The generator is encouraged to use latent codes to express information, meaning that the same method may be used with different generative models.

Code Link: https://github.com/gsu-kiranreddy/intro2DL-Repo2/tree/main/HW5