1. You observe a large gap between training and test performance. Which of these methods are commonly used to close such a gap? \*

|  |  |  |  |
| --- | --- | --- | --- |
| Sigmoid activation functions | **Drop-Out** | GANs | Skip connections |
|  | Using dropout on every iteration randomly some activations would be set as 0, which would create a sub-graph inside that large graph |  |  |

1. Which regularization methods lead to weight sparsity? Explain why\*

**L1 regularization leads to weight sparsity. While optimization, that is done based on the concept of gradient descent algorithm, L1 regularization makes the vector smaller, as most of its weights are zero. Geometrically, in L1 regularization, the viable solutions are limited to the corners, which leads to solutions to the optimization problem where many of the variables are zero.**

L1 regularization leads to weight sparsity. While optimization, that is done based on the concept of Gradient Descent algorithm, it is seen that if using L1 regularization, it brings sparsity to the weight vector by making smaller weights as zero. / make the vector smaller (sparse), as most of its components are useless (zero), the remaining non-zero components are very useful. In other words, in L1 regularization, the viable solutions are limited to the corners, leads to solutions to the optimization problem where many of the variables have value zero.

A vector or matrix with a maximum of its elements as zero is called a sparse vector or matrix.

Gradient Descent Algorithm: W\_t = W\_{t-1} – learning rate^\*(\partial L(w) / \partial (w))\_{w(t-1)}

1. Explain why an Autoencoder or Variational Autoencoder architecture can be used for pretraining parameters, and why this in some situations could lead to better performance? \*

**Autoencoder is an unsupervised neural network used for representation learning. Raw input is given to the encoder network, which transforms the data to a low-dimensional representation. The low-dimensional representation is then given to the decoder network, which tries to reconstruct the original input. They tried to make the output as close to the original input as possible. Autoencoder is often trained with a single layer encoder and a single layer decoder, to achieve better performance need to use many-layered encoders and decoders which is deep autoencoder. Deep autoencoder can exponentially reduce the computation, decrease the size of training data needs to learn, and experimentally, yield better compression. For deep autoencoder, pretraining approximates a good solution.**

**Same for Variational autoencoder, it learns to represent the input just in a compressed form as the latent space. The latent space can be considered as a set of distributions. Mathematically it is too computational cost to solve, so the solution is to build a parameterized model that can estimate the distribution.**

Autoencoder is an unsupervised neural network used for representation learning. It creates a low-dimensional representation of the original input data. Raw input is given to the encoder network, which transforms the data to a low-dimensional representation. The low-dimensional representation is then given to the decoder network, which tries to reconstruct the original input. They are trained by trying to make the reconstructed input form the decoder as close to the original input as possible.

Variational autoencoder learns to represent the input just in a compressed form called the latent space or the bottleneck. The latent space formed after training the model is not necessarily continuous and not easy to interpolate. The motivation behind expressing the latent attributes as a probability distribution can be very easily understood via statistical expressions. While estimating the distribution becomes impossible mathematically, a much simple and easier option is to build a parameterized model that can estimate the distribution.

1. Which of the following are common challenges with GANs?

**Training Instability**

Overfitting

**Vanishing gradients**

Uncalibrated probabilities

GAN (Generative Adversarial Network), Generative means the system will create content rather then classifying it. Adversarial means the system will train its components competitively. Network means the systems will be implemented as a neural network.

It consists of two main components, the generator and the discriminator. In principle, as the discriminator gets better at distinguishing real from generated images, the generator must produce images that more convincingly blend into the dataset, which is the ultimate goal. In practice, it is hard to set up a GAN to reliably train.

1. Which types of networks are necessary components of a GAN architecture?

**A Discriminator Network**

A Feed-Forward Network

A Cross-Stitch Network

**A Generator Network**

VAE is simply the generalization of an AE to a generative model[[1]](https://www.quora.com/Whats-the-difference-between-a-Variational-Autoencoder-VAE-and-an-Autoencoder#kUwsO). Basically, an AE will learn the latent structure of a feature set, whereas a VAE will learn to *generate* examples in this latent space.

VAE are generative autoencoders, meaning they can generate new instances that look similar to original dataset used for training.

VAE learns probability distribution of the data whereas autoencoders learns a function to map each input to a number and decoder learns the reverse mapping.

**Variational autoencoder is the generalization of autoencoder to a generative model. Autoencoder learns the latent structure of a feature set, where variational autoencoder learns to generate new examples in the latent space that look similar to the original input used for training.**

1. Mention three pre-training objectives, e.g., objectives for non-contrastive or contrastive self-supervised learning?

ELMO, BERT, SimCLR, Moco

Contrastive learning: SimCLR, Moco, PIRL, CCOP

[DeepClustering](https://openaccess.thecvf.com/content_ECCV_2018/html/Mathilde_Caron_Deep_Clustering_for_ECCV_2018_paper.html)  
[SeLA](https://arxiv.org/pdf/1911.05371.pdf)  
[SCAN](https://arxiv.org/abs/2005.12320)  
[SwAV](https://arxiv.org/abs/2006.09882)

CERT: CoLA, RTE, QNLI

Non-contrastive learning: BYOL, Simsiam, DirectPred

RBM models. GAN

1. Bonus Question: Explain why Multi-Task Neural Networks are Representation Learners?

The goal of an auxiliary task in Multi-Task Neural Network is to enable the model to learn representations that are shared or helpful for main task. All auxiliary tasks discussed so far do this implicitly. They are closely related to the main task, so that learning them likely allows the model to learn beneficial representations. A more explicit modelling is possible, for instance by employing a task that is known to enable a model to learn transferable representations. An autoencoder objective can also be used as an auxiliary task.

Multitask neural network results in improved learning efficiency and prediction accuracy when compared to training models separately. It does this by enabling the model to learn representations that are shared or helpful for main task. So what is learned for each task can help other tasks be learned better.