

Counterfactual Autoencoder for Unsupervised Semantic Learning

Saad Sadiq¹, Mei-Ling Shyu¹, Daniel Feaster²

¹Department of Electrical and Computer Engineering
University of Miami, Coral Gables, FL, USA

²Department of Public Health Sciences, Miller School of Medicine
University of Miami, Miami, FL, USA

Emails: ¹{s.sadiq, shyu}@miami.edu, ²dfeaster@med.miami.edu

Abstract

Deep Neural Networks (DNNs) are best known for being the state-of-the-art in Artificial Intelligence (AI) applications including natural language processing (NLP), speech processing, computer vision, etc. In spite of all recent achievements of deep learning, it has yet to achieve semantic learning required to reason about the data. This lack of reasoning is partially imputed to the boorish memorization of patterns and curves from millions of training samples and ignoring the spatiotemporal relationships. The proposed framework puts forward a novel approach based on variational autoencoders (VAEs) by using the potential outcomes model and developing the counterfactual autoencoders. The proposed framework transforms any sort of multimedia input distributions to a meaningful latent space while giving more control over how the latent space is created. This allows us to model data that is better suited to answer inference-based queries, which is very valuable in reasoning-based AI applications.

Keywords: Variational autoencoder (VAE), Variational inferencing, Counterfactual machines, Amortization, Gaussian processes

1. Introduction

Often in real-world applications such as multimedia, NLP, and medicine, large quantities of unlabeled data are generated every day. This surge in data gives rise to the challenging semantic gap problem (Lin, Shyu & Chen, 2012; Chen, Lin & Shyu, 2012; Zhu & Shyu, 2015; Sadiq, Yan, Shyu, Chen & Ishwaran, 2016) which is to reduce the gap between high level semantic concepts and their low level features (Sadiq, Tao, Yan & Shyu, 2017a; Sadiq, Zmieva, Shyu & Chen, 2018; Yan, Chen, Sadiq & Shyu, 2017). Despite rigorous research endeavors, this remains one of the most challenging problems in information sciences where we have overwhelming quantities of all sorts of fast, complex, heterogeneous and unstructured data. To handle such data, conventionally we have been utilizing descriptive models that try to find deterministic features and build probabilistic models (Chen & Kashyap, 1997; Chen, 2010; Lin, Shyu & Chen, 2013; Sadiq et al., 2017b). However, the problem with discriminative models is that they, generally, estimate a hyperplane. For example, when categorizing images of cats and dogs, data points on one side of the hyperplane are categorized as cats and everything on the other side as dogs. Discriminative models follow a condition in logistic regression that relaxes the computation of a joint probability $p(x, y)$ to a conditional probability $p(y|x)$ which is much easier to calculate because it maps directly to the hyperplane that divides between two clusters. This problem gets exacerbated in deep learning models due to the increase in dimensionality and boorish memorization of patterns. Mere rotations or color-inversions in the trained images can easily

confound very deep neural networks even though the modified images share the same semantics and structures as the original images (Hosseini & Poovendran, 2017).

Recently, there has been a growing concern of this shortcoming, resulting in a soaring interest in Knowledge Representation and Reasoning (KRR) (Liu, 2017) and reasoning based deep learning (Andreas, Rohrbach, Darrell & Klein, 2016; Santoro et al., 2017). We believe that the next generation of AI systems will need to have the ability to understand problems at a deeper level rather than just based on memorization of data. However, discriminative models fail completely in inferencing problems because they do not capture the underlying relationships in the input space. Figure 1 shows how the addition of an adversarial noise can completely fool the classification while the L2 loss between the original and modified images was minimal. The discriminative models considered them as identical images while producing completely different results. All images were classified with 99.9% confidence.

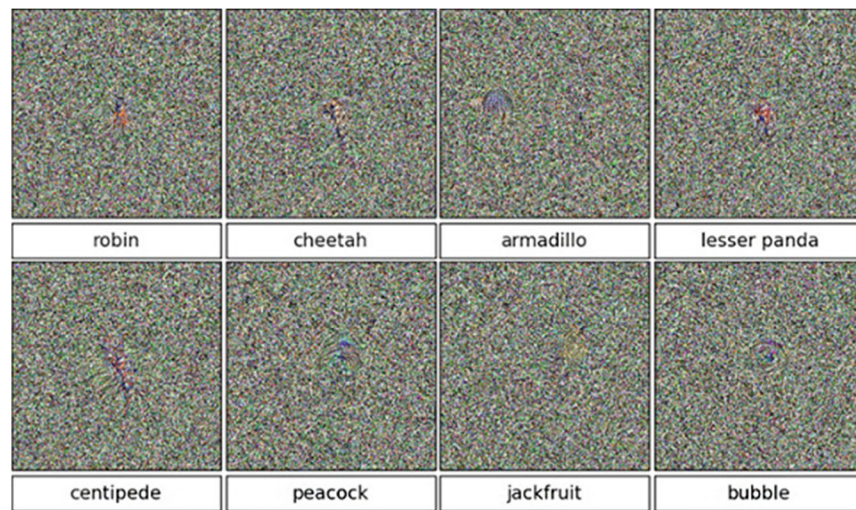


Figure 1. Classification results produced by state-of-the-art deep learning models, but fooled by simple noising up of the images (Nguyen, Yosinski & Clune, 2015)

There has been a growing interest to uncover such deep representations by mapping the observed data to latent spaces (Blei, Ng & Jordan, 2003; Neal, 2000; Roweis, 2003; Lei & Rinaldo, 2015; Koren, Bell & Volinsky, 2009). Thus, for reasoning applications, we have to look into generative models such as Variational Autoencoders (VAEs) (Kingma, Mohamed, Rezende & Welling, 2014) and Generative Adversarial Networks (GANs) (Goodfellow et al., 2014). This paradigm shift in training machine learning methods uses latent relationships and generative models and has proven to be successful in inferring relationships (Makhzani, Shlens, Jaitly, Goodfellow & Frey, 2015; Wetzl, 2017). Inference based training can open access to some interesting AI reasoning applications including: (1) Aiding users in understanding their surroundings or social media content (AI: ‘Mike just sent you a picture from his vacation in Thailand’; Human: ‘Great, is he at the beach?’; AI: ‘No, on a mountain’). (2) Aiding in decision making based on surveillance data (Human: ‘Did anyone enter this room last week?’; AI: ‘Yes, several instances found’; Human: ‘Were any of them carrying a black bag?’). (3) Health and safety applications (Human: ‘Can you see the baby in the baby monitor?’; AI: ‘Yes, I can’; Human: ‘Is he sleeping or playing?’). (4) Search and rescue missions (Human: ‘Is there smoke in any room around you?’; AI: ‘Yes, in room B6’; Human: ‘Look for people in room B6’).

Extending our previous work in semantic representation (Chen, Shyu & Kashyap, 2000; Chen, Zhu, Lin & Shyu, 2013; Yan, M. Chen, Shyu & Chen, 2015; Zhu, Lin, Shyu & Chen, 2011; Li, Chen, Shyu & Furht, 2002; Chen, Sista, Shyu & Kashyap, 1999; Chen, Rubin, Shyu & Zhang, 2006; Chen, Shyu, Zhang & Kashyap, 2001), here we propose a generative model that becomes the source for multimedia based inferencing frameworks. The proposed model implies the structure on the latent space in generative models and can be used for clustering, latent space arithmetic, semantic hashing and dimensionality reduction to make information retrieval faster. We can also use the proposed generative VAE with semi supervised learning to get very close to the state-of-the-art performance. Similar methods such as in (Makhzani et al., 2015) reported with good performance with MNIST data while using 10 labels instead of 5000 labels.

The rest of the paper is organized as follows, Section 2 describes the previous work done in generative models. The foundations of VAEs and statistical inferencing are presented in Section 3 to get an understanding of knowledge representation with the generative models. Section 4 provides the details about the proposed framework and performs the experiments in Section 5. Section 6 concludes the findings of this paper and identifies the future opportunities to extend this work.

2. Previous Work

Without statistical inferencing, we are simply confined to the bounds of our data and cannot infer new knowledge that extends beyond our data. Statistical inferencing in the reasoning paradigm is the process of generating conclusions about a population from a noisy and complicated sample. Until recently, deep generative models, such as Restricted Boltzmann Machines (RBMs), Deep Belief Networks (DBNs) and Deep Boltzmann Machines (DBMs) were trained primarily by Markov Chain Monte Carlo (MCMC)-based algorithms (Hinton, Osindero & Teh, 2006; Salakhutdinov & Hinton, 2009). In these approaches the MCMC methods compute the gradient of log-likelihood which becomes more imprecise as training progresses. This is because samples from the Markov Chains are unable to mix between modes fast enough. In recent years, generative models have been developed that may be trained via direct back-propagation and avoid the difficulties that come with MCMC training.

Some notable mentions of the existing methods include Knowledge-Based Artificial Neural Network (KBANN) (Towell, 1990), a hybrid learning method that leverages domain-knowledge and maps reformulated knowledge theories into neural networks. KBANNs proclaim promising results on biological applications with shallow encoding rules. Others have combined neural networks with Markov logic networks by learning the weights of the probabilistic logic clauses (Lippi & Frasconi, 2009). This technique requires structured equation modeling of background knowledge (e.g., bioinformatics and time-series forecasting) that suffers from the assumptions of model correctness and feature distributions. Bayesian Deep Learning (BDL) (Wan & Yeung, 2016) improves not only the perception tasks such as understanding of content (e.g., from text or image) but also the inference/reasoning tasks using principled probabilistic frameworks. However, BDL necessitates prior distribution information and assumes Gaussianity that restricts the input datasets and consequently restricts the applications. GANs (Radford & Chintala, 2015) are able to train the Question-Answering (QA) models but still lack the relationships between the black box features. Other noteworthy mentions include Memory Networks by Facebook AI Research (Sukhbaatar, Weston & Fergus, 2015), Neural Turing Machines by Google DeepMind (Graves, Wayne & Danihelka, 2014), and Watson by IBM (Gliozzo, Biran, Patwardhan &

McKeown, 2013). These large architectures do promise the reasoning capability, but they are proprietary systems that are not intended for public release. Our primary target is to propose a novel generalized open-source framework that imbeds the spatio-temporal data collinearities in the neural networks and activates the data flow invariant to sparsity and imbalance in the data. There are several other methods offering improvements on various dimensions such as the rate-distortion theory perspective presented in β -VAEs (Burgess et al., 2018). Their approach represents variational factors of data under a modified Evidence Lower Bound (ELBO) bound by a β condition. It creates the disentangled representations of the latent variables without a considerable loss in the reconstruction accuracy.

Another variant of the VAEs was proposed in (Dilokthanakul, 2016), where a Gaussian mixture was used as the prior distribution. This is helpful in performing uncluttered unsupervised clustering without over-regularization of the generative model. Walker, Doersch, Gupta & Hebert (2016) proposed a pixel-level dense trajectory prediction of given data frames. These frames were taken from a moving video scene where the algorithm estimates the changes over the time duration of one second using a conditional variational autoencoder setup. Another interesting approach was proposed by Sønderby, Raiko, Maaløe, Sønderby & Winther (2016) where a recursive inference model called the Ladder Variational Autoencoder was proposed. Their model corrects the generative distribution by using likelihood approximations. Their predictive log-likelihood claims superiority over the bottom-up inference in conventional VAEs.

3. Variational Autoencoders

Variational autoencoders (VAEs) are related to an unsupervised learning model called autoencoders. VAEs are used to learn a lower-dimensional feature representation from the unlabeled training data. The input data \mathbf{x} is converted to a latent space vector \mathbf{z} by a mapping function $p(\mathbf{z}|\mathbf{x})$. The encoder can take many forms, but neural networks usually outperform all other methods in learning the complex mapping functions. The hidden space \mathbf{z} is usually smaller than \mathbf{x} to avoid trivial solutions and serves as a form of dimensionality reduction as well. Then, \mathbf{z} represents the most important features in \mathbf{x} that can capture meaningful factors of the variation in data. The latent feature representation \mathbf{z} is utilized to reconstruct the original data by decoding them using an identical network to the encoder called the decoder to get the same dimensionality as \mathbf{x} , thus the term autoencoder – encoding itself. The data likelihood $p(\mathbf{x})$ is defined as taking the expectation over all possible values of \mathbf{z} , which is continuous, and the expression with the latent space \mathbf{z} can be obtained.

$$p_{\theta}(\mathbf{x}) = \int p_{\theta}(\mathbf{z}) p_{\theta}(\mathbf{x}|\mathbf{z}) dz \quad (1)$$

However, we are unable to take the gradient and maximize the likelihood $p(\mathbf{x}|\mathbf{z})$ for every possible value of \mathbf{z} because the integral is intractable. Here, $p_{\theta}(\mathbf{z})$ is a simple Gaussian prior, $p_{\theta}(\mathbf{x}|\mathbf{z})$ is a decoder neural network, and θ are the distribution parameters of the encoder and decoder. Similarly, the posterior density also becomes intractable due to the intractable data likelihood $p_{\theta}(\mathbf{x})$ in the denominator.

$$p_{\theta}(\mathbf{z}|\mathbf{x}) = \frac{p_{\theta}(\mathbf{x}|\mathbf{z})p_{\theta}(\mathbf{z})}{p_{\theta}(\mathbf{x})} \quad (2)$$

The solution that will enable us to learn this model is to define an additional encoder network $q_\phi(\mathbf{z})$, with a different set of parameters ϕ , that approximates $p_\theta(\mathbf{x}|\mathbf{z})$, in addition to the decoder network modeling $p_\theta(\mathbf{x}|\mathbf{z})$. This allows us to derive a lower bound on the data likelihood that is tractable and can be optimized. Since we are modeling a probabilistic generation of data in variational autoencoders, the encoder and decoder networks are probabilistic. Our encoder network $q_\phi(\mathbf{z})$ with parameters ϕ will output a mean $\mu(\mathbf{z}|\mathbf{x})$ and a diagonal covariance $\Sigma(\mathbf{z}|\mathbf{x})$. This will be the direct output of our encoder network. A similar method can be performed for the decoder network $p_\theta(\hat{\mathbf{x}}|\mathbf{z})$ which is going to start from \mathbf{z} and outputs the mean $\mu(\hat{\mathbf{x}}|\mathbf{z})$ and diagonal covariance $\Sigma(\hat{\mathbf{x}}|\mathbf{z})$ as shown in Figure 2(a).

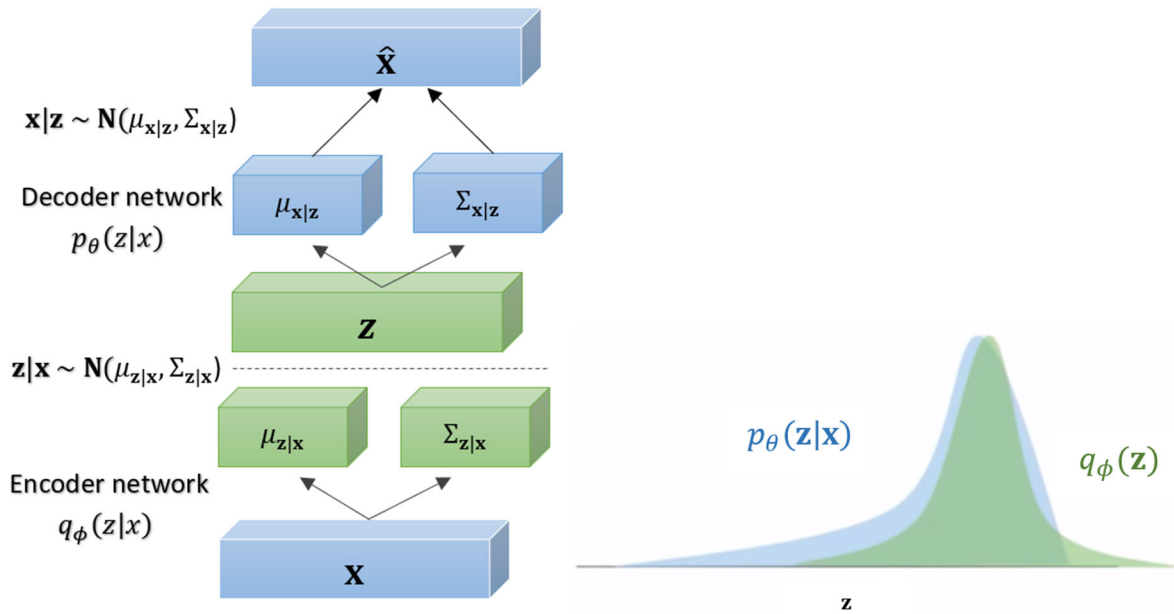


Figure 2. (a) Process of variational autoencoders where the latent space is sampled from a Gaussian distribution. (b) Process of bringing the mean and covariance of the latent space closer to $\sim N(0,1)$ by reducing the KL-Divergence or raising the ELBO.

3.1 KL-Divergence

It is important to note that dimensionality reduction by latent space representation works only if the inputs are highly correlated, for example, images from the same domain or texts from the same corpus. If completely random inputs are given each time while training an autoencoder, it will not be able to create a sound latent representation. This is because the encoded latent space will not cover the entire 2-D latent space and will have a lot of gaps in its output distribution. Hence, if we enter some values that the encoder has not fed to the decoder during the training phase, weird looking output images will be generated. This can be overcome by constraining the encoder output to have a random distribution, i.e., $\sim N(0,1)$, when producing the latent code. Thus, KL divergence is used as a weighted sum on some similarity measure to add the constraint on the latent space to match a normal distribution $\sim N(0,1)$ as shown in Figure 2(b).

$$D_{KL}(p(\mathbf{z}|\mathbf{x}) || q(\mathbf{z})) = \sum_i p(\mathbf{z}|\mathbf{x})_i \log \frac{p(\mathbf{z}|\mathbf{x})_i}{q(\mathbf{z})_i} \quad (3)$$

Here, $p(\mathbf{z})$ is the original latent distribution and $q(\mathbf{z})$ is the normal distribution $\sim N(0,1)$. Using variational inferencing, the Kullback-Leibler divergence can be reduced to the following

$$KL(q(\mathbf{z}) || p(\mathbf{z}|\mathbf{x})) = E_q \left(\log \left(\frac{q(\mathbf{z})}{p(\mathbf{z}|\mathbf{x})} \right) \right) \quad (4)$$

$$KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})) = E_q [\log q(\mathbf{z})] - E_q [\log p(\mathbf{z}|\mathbf{x})] \quad (5)$$

It is a distance measure between distributions (Hoffman, Blei, Wang & Paisley, 2013). However, $p(\mathbf{z}|\mathbf{x})$ cannot be calculated. Thus, applying the power of logarithms to Equation (5) transforms it into:

$$KL(q(\mathbf{z}) || p(\mathbf{z}|\mathbf{x})) = \underbrace{-(E_q(\log(p(\mathbf{x}, \mathbf{z}))) - E_q(\log(q(\mathbf{z}))))}_{\text{ELBO}} + \log p(\mathbf{x}) \quad (6)$$

The first two terms on the right hand side are equivalent to maximizing the ELBO, which only requires to calculate the joint distribution $p(\mathbf{x}, \mathbf{z})$. Please note that the third term does not involve q and thus this distribution can be ignored because it is just a number in optimizing the variational distribution. There is, however, always a trade-off between how accurately the latent space can fit into the desired Gaussian shape and how well informative new data can be generated.

3.2 Reparameterization

Autoencoders are generally simpler to train because we simply have to backpropagate the reconstruction loss across the weights of the network. However, VAEs are not as simple to optimize because the sampling operation is not differentiable. This means that the gradients from the reconstruction error cannot be propagated to the encoder. The reparameterization trick is to stretch the encoded standard deviation with an additional random noise as shown in Figure 3. This is equivalent to random but becomes a linear equation.

$$z^{(i)} = \mu^{(i)} + \sigma^{(i)} \odot \varepsilon_i \quad (7)$$

$$\varepsilon_i \sim N(0,1) \quad (8)$$

Here, i represents the batch and ε_i is the random noise that helps us backpropagate the network. With the reparametrized form, we changed $q(z|\theta, \mathbf{x}) \rightarrow q(\theta, \mathbf{x}, \varepsilon)$.

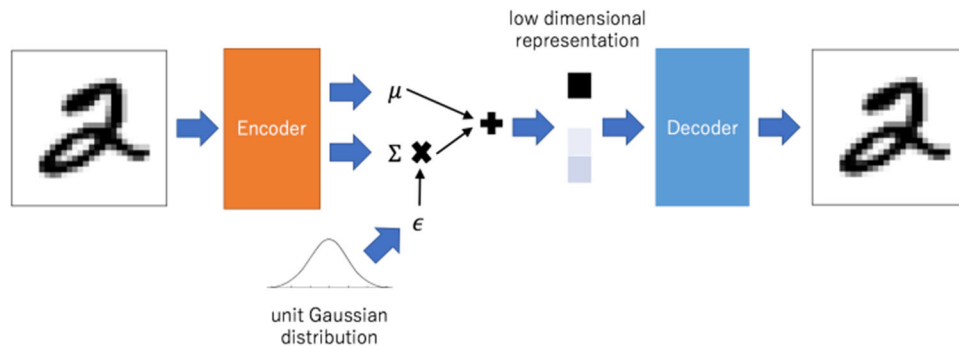


Figure 3. Complete process of a vanilla VAE using reparameterization

4. Framework

In this paper, a novel object correlation model that uses VAE for feature translation to the latent space is proposed (as illustrated in Figure 4). We start with our minibatch \mathbf{X}_i that is passed through the encoder network which can be a Convolutional, Bi-LSTM or a fully-connected network depending on the data type. Two vector outputs μ_θ and Σ_θ (i.e., the mean and the covariance matrix) can be obtained. After applying the reparameterization trick, we sample the actual latent stochastic representation of the image $\mathbf{z}|\mathbf{x}$. The K dimensional latent encoding is decoded using the decoder network, which is similar to the encoder network, to get the replica of our minibatch \mathbf{X}_i . The network trains in three steps. The first part is the reconstruction phase where we train the VAE to reproduce the minibatch \mathbf{X}_i using the L2 loss and Adam optimizer. The K -dimensional latent vector obtained from the first phase is called \mathbf{z}_C . The second phase is to train and produce the counterfactual latent representation denoted by \mathbf{z}_T . The counterfactual latent encodings are produced by our proposed Shallow Randomized Backpropagation (ShRB) networks. The ShRB networks, explained in the next subsection, allows us to produce Gaussian versions of the original latent encoding with the capability to manipulate the latent space. The final step is to backpropagate on the loss of these two counterfactual terms and train the encoder to minimize the KL divergence between $\mathbf{z}_C || \mathbf{z}_T$. There will be two losses. The first one is the reconstruction loss to train the horizontal pass of the variational autoencoder; while the second one is the KL loss that will help train the ShRB networks.

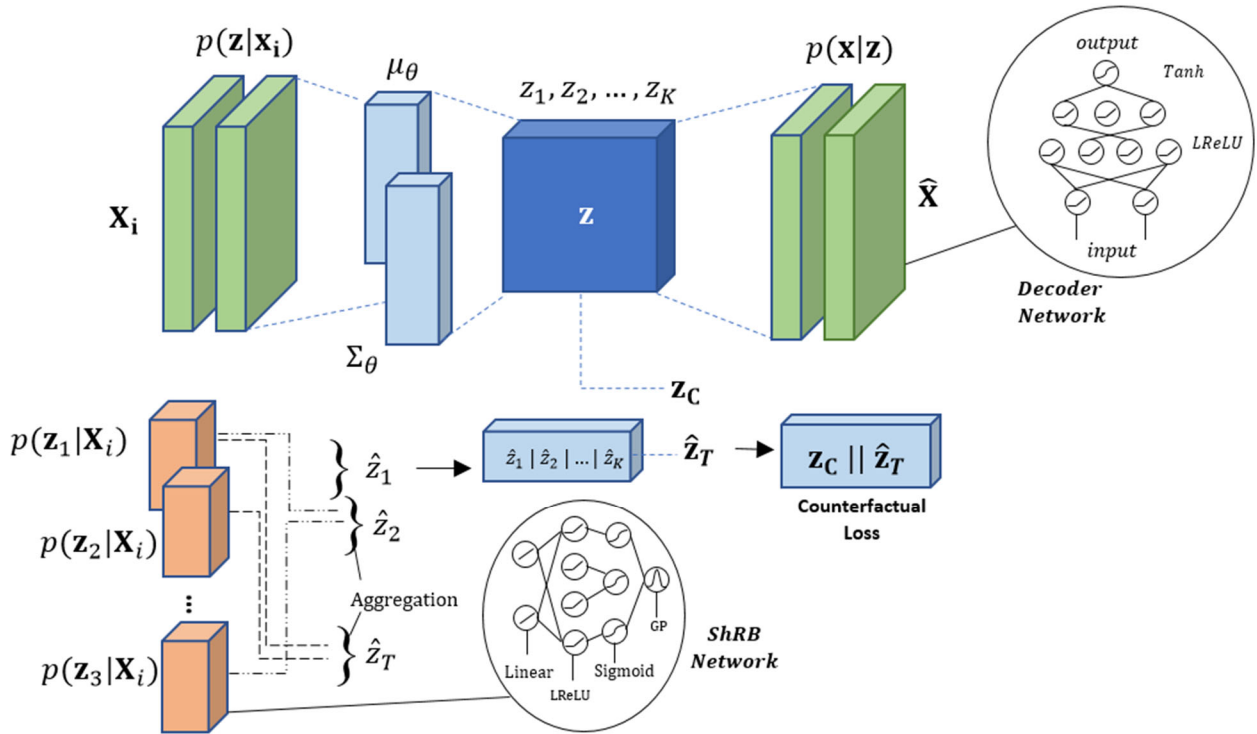


Figure 4. The proposed counterfactual autoencoder framework

4.1 Shallow Randomized Backpropagation (ShRB)

It is interesting to note that VAEs are very bad at disentangling and separating the clusters or subgroups of data in the hidden space. Thus, we propose an unsupervised subgrouping method, called Shallow Randomized Backpropagation (ShRB) networks, which completely removes the subsampling / pooling layers and maximizes the diversity of the hidden space variables. The ShRB networks primarily takes the latent space created by VAEs and additionally separates the variance of the feature covariance matrix by sampling the features of the minibatch. The ShRB networks create disjoint Gaussians. That is, each latent dimension $z_k \in K$ is trained on each ShRB network.

Suppose that each ShRB network is represented by the function $\hat{f}(\cdot)$, and z_k is the univariate continuous valued outcome vector, and ε is the error term. Then for an i^{th} input batch \mathbf{X}_i and the latent vector z , the goal is to closely predict the latent vector z_k .

If $\hat{f}(\mathbf{X})$ predicts $z = f_0(\mathbf{X}) + \varepsilon$, where $\hat{f}(\mathbf{X})$ is the predicted z , and $f_0(\cdot)$ is the learned functions, then

$$PE(f) = \mathbb{E}_{\mathcal{L}} \mathbb{E}_{\mathbf{X}, z} [z - \hat{f}(\mathbf{X})]^2 \quad (9)$$

$$= \sigma^2 + \mathbb{E}_{\mathbf{X}} [\hat{\mu}(\mathbf{X}) - f_0(\mathbf{X})]^2 \quad \} \text{Bias.}$$

$$+ \mathbb{E}_{\mathcal{L}, \mathbf{X}} [\hat{\mu}(\mathbf{X}) - \hat{f}(\mathbf{X})]^2 \quad \} \text{Variance} \quad (10)$$

Here, $PE()$ is the prediction error, $\mathbb{E}_{\mathcal{L}}$ is the expectation over the learning dataset, $\mathbb{E}_{\mathbf{X}, z}$ represents the expectation over the test dataset, and $\hat{\mu}(\mathbf{X}) = \mathbb{E}_{\mathcal{L}}[\hat{f}(\mathbf{X})]$. The first term σ^2 is the internal noise and this is the lower bound on the generalization error. The second term $\mathbb{E}_{\mathbf{X}} [\hat{\mu}(\mathbf{X}) - f_0(\mathbf{X})]^2$ is the bias, described as the difference between the true predictor and the mean of the learned predictor. The final term is the variance $\mathbb{E}_{\mathcal{L}, \mathbf{X}} [\hat{\mu}(\mathbf{X}) - \hat{f}(\mathbf{X})]^2$, described by the difference between the predictor and its mean value. It is basically a mean-squared error decomposition. There are two observations from this. The prediction error for each predictor is always bigger than the averaged predictor and the amount is equal to the variance. Thus by taking the average, we can completely remove the variance, and it drops to zero. Moreover, if our original predictor is unbiased, then taking the average of it will also be unbiased. Hence, it is in some ways a very effective predictor, taking an average of an unbiased high variance predictor and turns it into a stable and accurate ensemble learner. In cases where we have high dimensional latent spaces, then the perturbation would not be enough to decorrelate the predictors and to reduce the variance. Thus, we further sample from the feature set and pass only 1/3rd of the features for every sample. This is another form of randomization in the process to reduce the variance.

4.2 Tensor Slicing and Conversion to Gaussian

We assume that z_1, \dots, z_K are all independent latent vectors, thus giving us only a diagonal variance matrix. Instead of calculating $O(n^2)$ computations, we trade the complexity for the bias and calculate only $O(n)$ steps through variational approximation. This is done by slicing the latent tensors and estimating them individually through the ShRB networks. By this approach, we know that we do not estimate the true Gaussian nature of the latent tensor, but we will get the

best answer under the complexity constraints that the application can tolerate. The tensors need to be sliced along each dimension of the latent vector \mathbf{z} such that, given the vector $\mathbf{z} \in R_{1 \times K}$ the notation $z_k \in R_K$ denotes the k-th dimension of \mathbf{z} . Each predicted z_k is scaled between 0 and 1 values using sigmoid activation and converted to Gaussian using probabilities to z-score transformation activation.

$$\int_{-\infty}^z e^{-(x)^2} dx = \text{sigmoid}(p) \quad (11)$$

$$0.5 + \int_0^z e^{-t^2} dt = 0.5 + \frac{1}{2} \sqrt{\pi} \text{erf}(z) \quad (12)$$

4.3 Counterfactual Approach

Conventionally in VAEs, almost all methods attempt to force the latent variable to match a normal distribution $\sim N(0,1)$. This approach forces a distribution that brings the latent vector mean towards zero, imbalancing the underlying data relationships such that, the outliers and imbalanced data points are also conditioned to be generated under a zero mean restriction. Our proposed counterfactual approach takes the original latent variable distribution as a prior and generates a counterfactual Gaussian of the latent variable \mathbf{z} without forcing any prior conditions on it. This allows us to learn a latent distribution that allows better disentanglement between each latent vector along all dimensions. Formally, let $\{(\mathbf{x}_1, \mathbf{z}_1), \dots, (\mathbf{x}_i, \mathbf{z}_i)\}$ denote the minibatches where \mathbf{x}_i is a minibatch and \mathbf{z}_i represents the latent vector for that minibatch. After learning the latent representation, we call the non-transformed \mathbf{z}_i as $\mathbf{z}_{i,C}$ (where C indicates no transformation) and the ShRB estimated latent vector $\hat{\mathbf{z}}_i$ is denoted by $\hat{\mathbf{z}}_{i,T}$ (where T represents being transformed). Our goal is to train on the loss incurred by the differences between $\mathbf{z}_{i,C}$ and $\hat{\mathbf{z}}_{i,T}$. The loss is defined as

$$\tau(\mathbf{x}) = \mathbb{E}[\mathbf{z}_{i,T} | \mathbf{x}_i = \mathbf{x}] - \mathbb{E}[\mathbf{z}_{i,C} | \mathbf{x}_i = \mathbf{x}] \quad (13)$$

Equation (12) relies on what is so-called the counterfactual framework or the potential outcomes model (Neyman, Dabrowska & Speed, 1990). In this framework, one models what an individual vector would look like if it were generated as a Gaussian although the observed vector is non-Gaussian. This modification allows us to use a broader set of distributions as priors for the latent code as follows

1. Preserves the distribution giving better manifold;
2. Removes the gaps between the latent representations;
3. We can switch to the variance as a loss function and get the maximized separation;
4. It does not matter how high dimensional we treat each z_k individually; and
5. It can be scaled without affecting the Gaussian conversion.

5. Experiments and Results

In this section, we perform several experiments using the proposed Counterfactual Autoencoders and the MNIST and Fashion MNIST datasets. One of the advantages of using the ShRB networks is that they create disjoint estimates of the sliced latent dimensions. For example, in any given image, some ShRB networks will see the top part of an image while other ShRB networks will see the bottom part of it. Based on what each ShRB network sees, it will create a

different set of μ and σ estimates for each $z_k \in K$. This increases variability in the \mathbf{z} estimates and gives us the benefit of scalability and the ability to manipulate the z-space easily.

5.1 Performance Comparison

The experiment was compared with the original VAE (Kingma et al., 2014) and a comparative approach called the Maximum-mean discrepancy VAEs (Zhao, Song & Ermon, 2017). The results of the comparison are shown as below

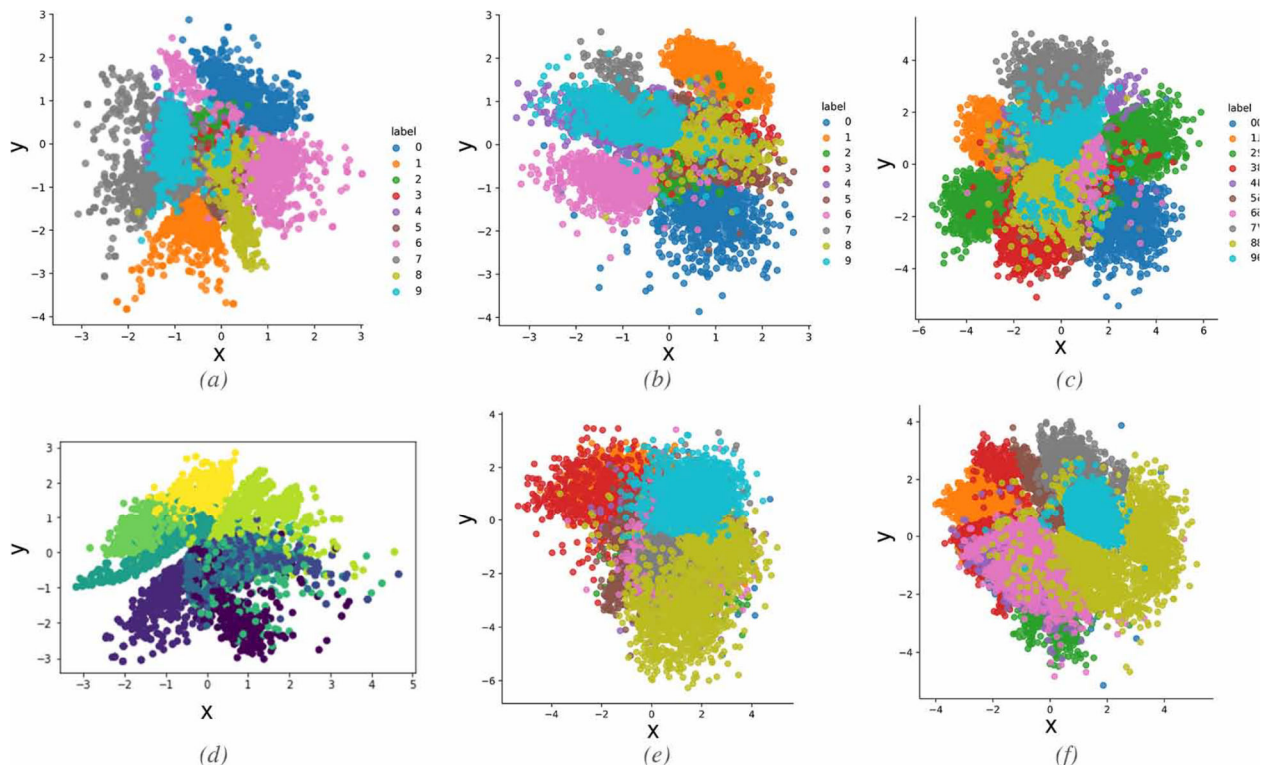


Figure 5: The learned latent space manifold for MNIST dataset using (a) MMD VAE (b) Vanilla VAE and (c) the Proposed Counterfactual Autoencoder. Also illustrated are the learned manifold for Fashion MNIST data compared between using (d) MMD VAE (e) Vanilla VAE and (f) the proposed Counterfactual Autoencoder

While comparing the learned manifold spaces, we see a near circular 2-D Gaussian shape with good disentanglement achieved by the proposed counterfactual autoencoder in Figure 5(c). The learned manifold depicts sharp transitions between data points indicating that the coding space is filled and contains no gaps. By contrast, Figures 5(a) and 5(b) illustrate the coding space of the MMD and vanilla VAE with almost similar architectures to that of the proposed counterfactual autoencoder. However, it can be observed that the two methods roughly match the shape of a 2-D Gaussian distribution and there are several gaps in the manifold. These points map to no data and end up creating garbage data. The methods were not able to capture the manifold as good as the counterfactual autoencoder. Similar observation is found when comparing the three methods when trained on the Fashion MNIST dataset as illustrated in Figures 5 (d), (e) and (f). The Fashion MNIST dataset is more ambiguous than MNIST, thus we see bizarre distributions for

MMD and vanilla VAE. Whereas the proposed counterfactual autoencoders achieves perfect 2D Gaussian shape with good disentanglement between classes.

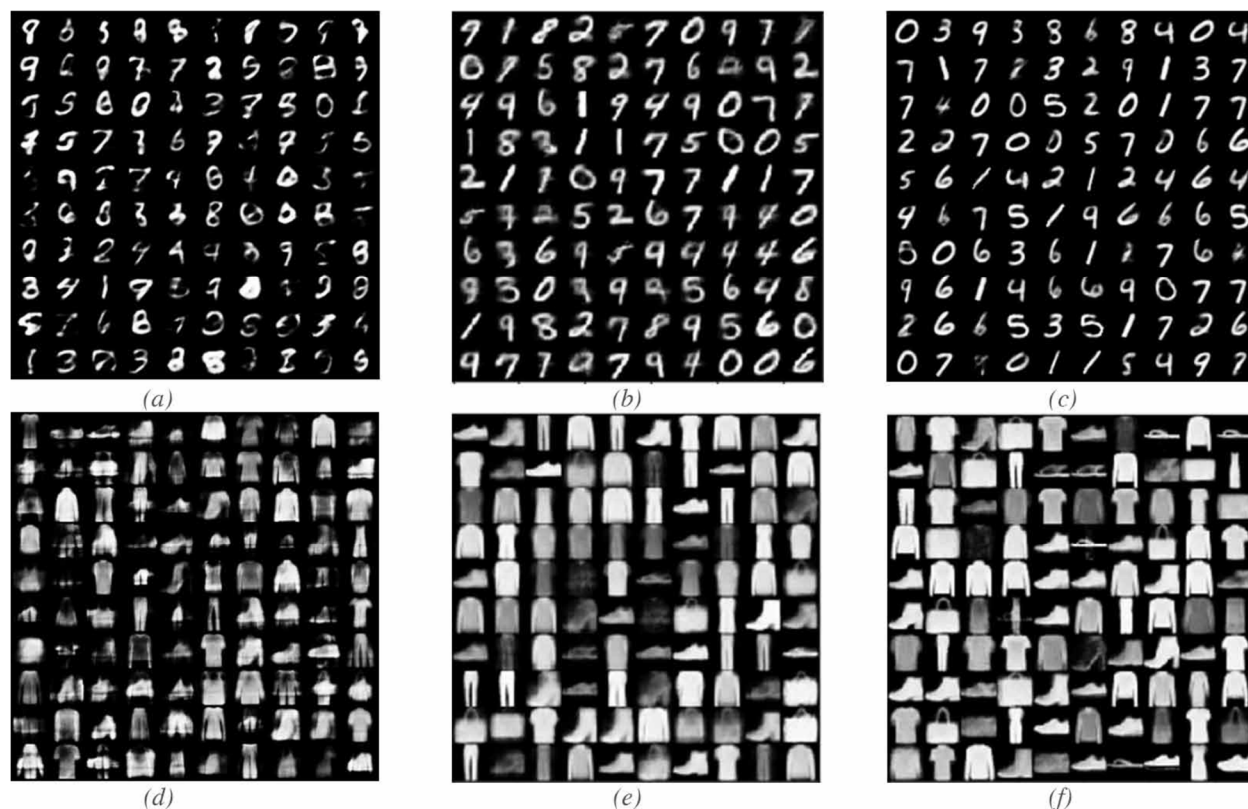


Figure 6. Random data generation between the conventional VAEs, MMD VAEs and the proposed Counterfactual autoencoder. All methods were allowed to train up to 20 epochs and then the results were sampled from the trained decoders for MNIST (a) using vanilla VAEs, (b) using MMD VAEs and (c) the proposed Counterfactual Autoencoder. Similar samples were generated for the Fashion MNIST dataset trained on (d) vanilla VAE, (e) MMD VAE and (f) the Proposed Counterfactual Autoencoder.

To evaluate the generative capability of the three models, it can be observed that the proposed counterfactual autoencoders generate clear and disentangled data over the entire latent space. The other two compared methods either produce incomplete or mixed representations that are cluttered together. We only trained the models with only 2-D latent spaces for observational purposes, but the model is scalable to any number of higher dimension latent spaces.

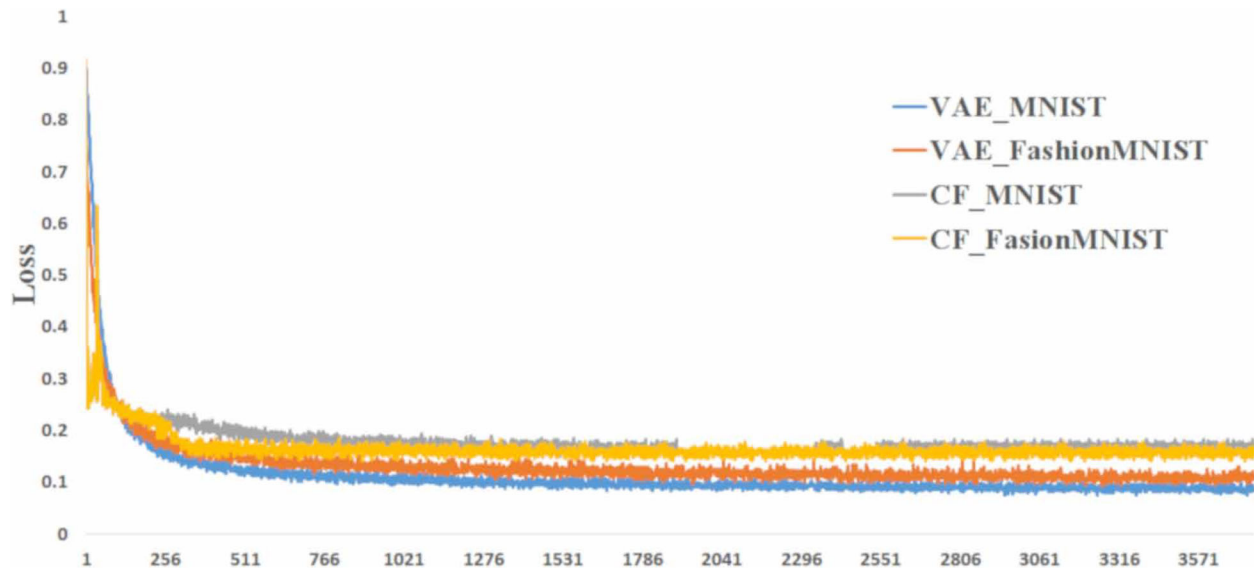


Figure 7. The training loss as compared between the proposed Counterfactual-Autoencoder and vanilla VAE

Figure 7 illustrates the training loss between the proposed method and the conventional VAE over the MNIST data. It should be noted that the training loss is an indicator of the reconstruction loss only and does not translate to other aspects of the generative models. Although the training losses of vanilla VAEs were lower than the proposed counterfactual autoencoders (called CF in Figure 7), their latent spaces were very entangled and are not very useful for data generation.

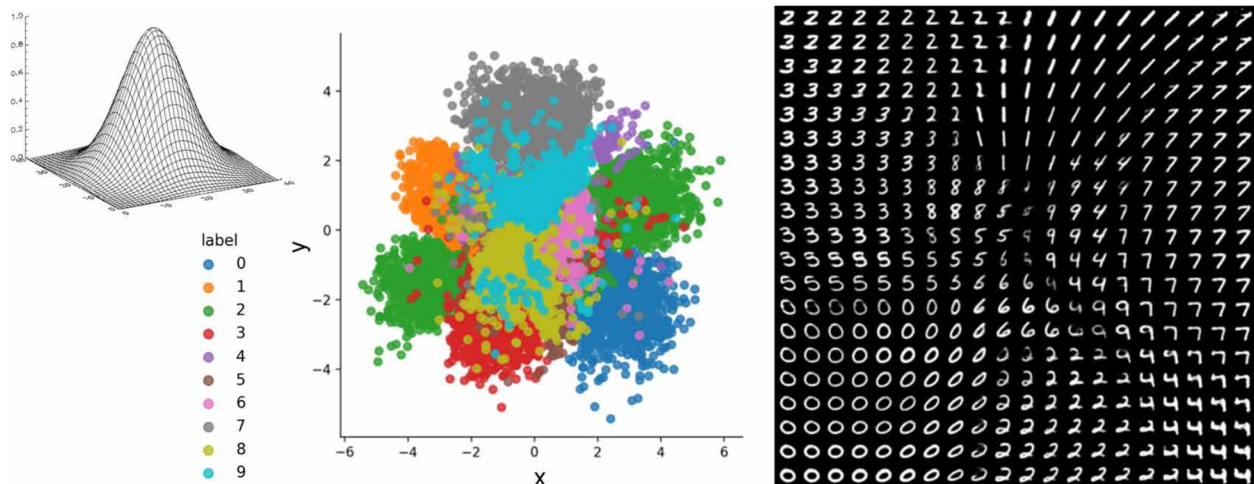


Figure 8. The hidden space manifold for MNIST data generated after 50 epochs

Figure 8 shows the learned manifold over the MNIST data for the proposed method. Images were generated by uniformly sampling the Gaussian percentiles along each hidden code dimension z in the 2-D Gaussian space. Sharp transitions in the coding space indicate that the images generated by interpolating within z lie on the data manifold with a good precision.

6. Conclusion and Future work

Deep Learning puts forward some of the most sophisticated machine learning solutions to the current state-of-art Artificial Intelligence (AI). However, these methods are based on a monolithic approach of being confined within the data. This results in an array of non-interpretable and high dimensional features that cannot be reasoned or used in any methodical framework. In spite of all the success and power of Deep Neural Networks (DNNs), AI is still far from how a human analyzes the data. The answer may lie in an inference-based approach that looks at the problem in a reduced heuristics space, like humans, and uses the power of the latent space arithmetic to put 2 and 2 together. Current reasoning frameworks are either too narrow or require too much human assistance that renders them un-scalable. In this paper a novel approach called counterfactual autoencoders is proposed that regularizes hidden space representations using randomized shallow learners and trains the generative network using a counterfactual loss. The proposed approach transforms the hidden representations to near perfect Gaussians with good disentanglement and provides wide-ranging control to manipulate and explore the latent space. This provides us with opportunities to use the Counterfactual Autoencoder in order to solve semantic learning, reasoning and inferencing problems. The algorithm was tested only on the prior mentioned two datasets, but the proposed framework's capability can be easily extrapolated to other multimedia datasets. As a future prospect, further specific use-cases will be explored and presented in upcoming publications.

7. References

- Andreas, J., Rohrbach, M., Darrell, T., & Klein, D. (2016). Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 39-48).
- Bigham, J. P., Jayant, C., Ji, H., Little, G., Miller, A., Miller, R. C., & Yeh, T. (2010). VizWiz: Nearly real-time answers to visual questions. In *Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology* (pp. 333-342).
- Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3, 993-1022.
- Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., & Lerchner, A. (2018). Understanding disentangling in β -VAE. arXiv preprint arXiv:1804.03599.
- Chen, S.-C. & Kashyap, R. L. (1997). Temporal and spatial semantic models for multimedia presentations. In 1997 International Symposium on Multimedia Information Processing (pp. 441-446).
- Chen, S.-C., Sista, S., Shyu, M.-L., & Kashyap, R. L. (1999). Augmented transition networks as video browsing models for multimedia databases and multimedia information systems. In *Proceedings of 11th IEEE International Conference on Tools with Artificial Intelligence* (pp. 175-182).
- Chen, S.-C., Shyu, M.-L., & Kashyap, R. L. (2000). Augmented transition network as a semantic model for video data. *International Journal of Networking and Information Systems*, 3(1), 9-25.
- Chen, S.-C., Shyu, M.-L., Zhang, C., & Kashyap, R. L. (2001). Identifying overlapped objects for video indexing and modeling in multimedia database systems. *International Journal on Artificial Intelligence Tools*, 10(4), 715-734.

- Chen, S.-C., Rubin, S. H., Shyu, M.-L., & Zhang, C. (2006). A dynamic user concept pattern learning framework for content-based image retrieval. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 36(6), 772-783.
- Chen, S.-C. (2010). Multimedia databases and data management: a survey. *International Journal of Multimedia Data Engineering and Management (IJMDEM)*, 1(1), 1-11.
- Chen, C., Lin, L., & Shyu, M.-L. (2012). Re-ranking algorithm for multimedia retrieval via utilization of inclusive and exclusive relationships between semantic concepts. *International Journal of Semantic Computing*, 6(2), 135-154.
- Chen, C., Zhu, Q., Lin, L., & Shyu, M.-L. (2013). Web media semantic concept retrieval via tag removal and model fusion. *ACM Transactions on Intelligent Systems and Technology*, 4(4), 61.
- Czerlinski, J., Gigerenzer, G., & Goldstein, D. G. (1999). How good are simple heuristics?. In *Simple Heuristics That Make Us Smart* (pp. 97-118). Oxford University Press.
- Dilokthanakul, N., Mediano, P. A., Garnelo, M., Lee, M. C., Salimbeni, H., Arulkumaran, K., & Shanahan, M. (2016). Deep unsupervised clustering with gaussian mixture variational autoencoders. arXiv preprint arXiv:1611.02648.
- Graves, A., Wayne, G., & Danihelka, I. (2014). Neural turing machines. CoRR abs/1410.5401. <http://arxiv.org/abs/1410.5401>
- Gliozzo, A., Biran, O., Patwardhan, S., & McKeown, K. (2013). Semantic technologies in IBM Watson. In *Proceedings of the Fourth Workshop on Teaching NLP and CL* (pp. 85-92).
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems* (pp. 2672-2680).
- Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7), 1527-1554.
- Hoffman, M. D., Blei, D. M., Wang, C., & Paisley, J. (2013). Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1), 1303-1347.
- Hosseini, H. & Poovendran, R. (2017). Deep neural networks do not recognize negative images. CoRR abs/1703.06857. <http://arxiv.org/abs/1703.06857>
- Kingma, D. P., Mohamed, S., Rezende, D. J., & Welling, M. (2014). Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems* (pp. 3581-3589).
- Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, (8), 30-37.
- Lei, J. & Rinaldo, A. (2015). Consistency of spectral clustering in stochastic block models. *The Annals of Statistics*, 43(1), 215-237.
- Li, X., Chen, S.-C., Shyu, M.-L., & Furht, B. (2002). Image retrieval by color, texture, and spatial information. In *Proceedings of the 8th International Conference on Distributed Multimedia Systems* (pp. 1-8).
- Lippi, M. & Frasconi, P. (2009). Prediction of protein β -residue contacts by Markov logic networks with grounding-specific weights. *Bioinformatics*, 25(18), 2326-2333.

- Lin, L., Shyu, M.-L., & Chen, S.-C. (2012). Association rule mining with a correlation-based interestingness measure for video semantic concept detection. *International Journal of Information and Decision Sciences*, 4(2-3), 199-216.
- Lin, L., Shyu, M.-L., & Chen, S.-C. (2013). Rule-based semantic concept classification from large-scale video collections. *International Journal of Multimedia Data Engineering and Management*, 4(1), 46-67.
- Liu, H. C., You, J. X., Li, Z., & Tian, G. (2017). Fuzzy Petri nets for knowledge representation and reasoning: A literature review. *Engineering Applications of Artificial Intelligence*, 60, 45-56.
- Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., & Frey, B. (2015). Adversarial autoencoders. arXiv preprint arXiv:1511.05644.
- Neal, R. M. (2000). Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9(2), 249-265.
- Nguyen, A., Yosinski, J., & Clune, J. (2015). Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 427-436).
- Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. CoRR abs/1511.06434. <http://arxiv.org/abs/1511.06434>
- Roweis, S. T. (2003). Factorial models and refiltering for speech separation and denoising. In *Proceedings of the Eighth European Conference on Speech Communication and Technology* (pp. 1009-1012).
- Sadiq, S., Yan, Y., Shyu, M.-L., Chen, S.-C., & Ishwaran, H. (2016). Enhancing multimedia imbalanced concept detection using VIMP in random forests. In *Proceedings of the IEEE 17th International Conference on Information Reuse and Integration* (pp. 601-608).
- Sadiq, S., Tao, Y., Yan, Y., & Shyu, M.-L. (2017a). Mining anomalies in Medicare big data using patient rule induction method. In *Proceedings of the IEEE Third International Conference on Multimedia Big Data* (pp. 185-192).
- Sadiq, S., Yan, Y., Taylor, A., Shyu, M.-L., Chen, S.-C., & Feaster, D. (2017b). AAFA: Associative affinity factor analysis for bot detection and stance classification in Twitter. In *Proceedings of the IEEE International Conference on Information Reuse and Integration* (pp. 356-365).
- Sadiq, S., Zmievá, M., Shyu, M.-L., & Chen, S.-C. (2018). Reduced residual nets (Red-Nets): Low powered adversarial outlier detectors. In *Proceedings of the IEEE International Conference on Information Reuse and Integration for Data Science* (pp. 436-443).
- Salakhutdinov, R. & Hinton, G. (2009). Semantic hashing. *International Journal of Approximate Reasoning*, 50(7), 969-978.
- Santoro, A., Raposo, D., Barrett, D. G., Malinowski, M., Pascanu, R., Battaglia, P., & Lillicrap, T. (2017). A simple neural network module for relational reasoning. In *Advances in Neural Information Processing Systems* (pp. 4967-4976). CoRR abs/1706.01427. <http://arxiv.org/abs/1706.01427>

- Sønderby, C. K., Raiko, T., Maaløe, L., Sønderby, S. K., & Winther, O. (2016). Ladder variational autoencoders. In *Advances in Neural Information Processing Systems* (pp. 3738-3746).
- Splawa-Neyman, J., Dabrowska, D. M., & Speed, T. P. (1990). On the application of probability theory to agricultural experiments. *Essay on principles*. Section 9. *Statistical Science*, 465-472.
- Sukhbaatar, S., Weston, J., & Fergus, R. (2015). End-to-end memory networks. In *Advances in Neural Information Processing Systems* (pp. 2440-2448).
- Todd, P. M. & Gigerenzer, G. (2000). Précis of simple heuristics that make us smart. *Behavioral and Brain Sciences*, 23(5), 727-741.
- Towell, G. G., Shavlik, J. W., & Noordewier, M. O. (1990). Refinement of approximate domain theories by knowledge-based neural networks. In *Proceedings of the Eighth National Conference on Artificial Intelligence* (Vol. 861866).
- Tversky, A. & Kahneman, D. (1974). Judgment under uncertainty: Heuristics and biases. *Science*, 185(4157), 1124-1131.
- Walker, J., Doersch, C., Gupta, A., & Hebert, M. (2016). An uncertain future: Forecasting from static images using variational autoencoders. In *European Conference on Computer Vision* (pp. 835-851). Springer, Cham.
- Wang, H. & Yeung, D. Y. (2016). Towards Bayesian deep learning: A framework and some existing methods. *IEEE Transactions on Knowledge and Data Engineering*, 28(12), 3395-3408.
- Wetzel, S. J. (2017). Unsupervised learning of phase transitions: From principal component analysis to variational autoencoders. *Physical Review E*, 96(2), 022140.
- Yan, Y., Chen, M., Shyu, M.-L., & Chen, S.-C. (2015). Deep learning for imbalanced multimedia data classification. In *Proceedings of the IEEE International Symposium on Multimedia* (pp. 483-488).
- Yan, Y., Chen, M., Sadiq, S., & Shyu, M.-L. (2017). Efficient imbalanced multimedia concept retrieval by deep learning on spark clusters. *International Journal of Multimedia Data Engineering and Management*, 8(1), 1-20.
- Zhao, S., Song, J., & Ermon, S. (2017). Infovae: Information maximizing variational autoencoders. arXiv preprint arXiv:1706.02262.
- Zhu, Q., Lin, L., Shyu, M.-L., & Chen, S.-C. (2011). Effective supervised discretization for classification based on correlation maximization. In *Proceedings of the IEEE International Conference on Information Reuse and Integration* (pp. 390-395).
- Zhu, Q. & Shyu, M.-L. (2015). Sparse linear integration of content and context modalities for semantic concept retrieval. *IEEE Transactions on Emerging Topics in Computing*, 3(2), 152-160.