Lecture 2 - Exercise sheet

Master in Deep Learning - Generative Models

Pablo Miralles-González, Javier Huertas-Tato

January 15, 2025

Problem 1.

What happens when you adjust the hyperparameter β that multiplies the KL divergence in the β-VAE loss? Explain with your own words.

Solution: In a β-VAE, the objective function is modified to balance the reconstruction loss and the KL divergence term with a hyperparameter β:

- **Small** β: When $\beta < 1$, the model prioritizes reconstruction accuracy over learning a continuous distribution that is similar to the prior distribution we select (typically a standard normal distribution, that is, one with zero mean and unit standard deviation, as seen in class). This results in highly detailed reconstructions but a latent space with a discrete distribution that cannot be used for generation or manipulation of examples.
- **Large** β: When $\beta > 1$, the KL divergence term is weighted more heavily, encouraging the latent space to closely resemble a standard normal distribution, but at the cost of poorer reconstruction quality. This has been argued to lead to more disentangled representations, where individual latent dimensions capture independent generative factors.

Problem 2.

How would you discover which vectors represent specific features in the latent space? For example, if you trained a VAE on MNIST, how would you discover which vector corresponded on average to a specific digit?

Solution: Let us consider the example of the digit 7.

1. Take some samples labeled with the digit 7.
2. Apply the encoder network to these samples, obtaining their latent means $\mu_1, \mu_2, \ldots, \mu_n$.
3. Obtain the average of the means $\hat{\mu} = \sum \mu_i$.

This vector $\hat{\mu}$ represents the centroid of the class "7" in the latent space, the perfect instance of a handwritten 7 digit if you wish. We can also take the standard deviations from the encoder and calculate the mean standard deviation from this class $\hat{\sigma} = \sum \sigma_i$. If we assume that the distribution of latent representations for digits 7 is also a normal distribution, then we can sample new 7s from a $\mathcal{N}(\hat{\mu}, \hat{\sigma})$.

Problem 3.

How would you discover vectors in the latent space that produce a specific change? For example, consider a VAE trained with images of faces. How would you discover a vector that, when summed to the latent representation of a face, makes that face smile? Assume that you have access to the feature for each of the training images. In the example, you know if a face in the image is smiling or not.

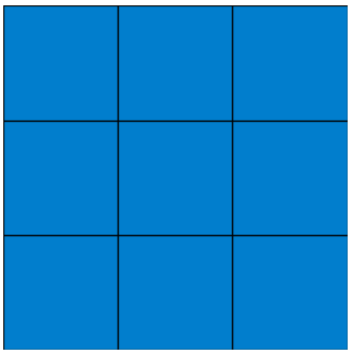


Figure 1: Original input image.

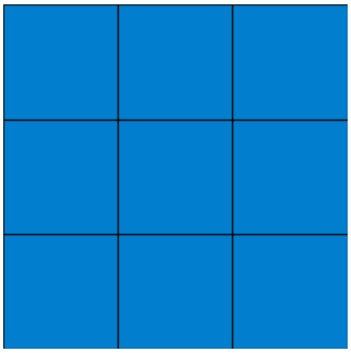
Solution:

1. **Obtain latent representations**: Pass both smiling and non-smiling images through the encoder to obtain their latent means, as in the previous exercise, obtaining one centroid for each class: μ_{smiling} and $\mu_{\text{not smiling}}$.
2. **Compute Difference Vector**: Take the mean vectors of smiling and non-smiling faces and subtract them:
$$\mu_{\text{make smile}} = \mu_{\text{smiling}} - \mu_{\text{not smiling}}$$
3. **Test the Vector**: Add $v_{\text{make smile}}$ to the latent representations of other faces that do not smile and apply the decoder to them to check if the resulting images have smiles.

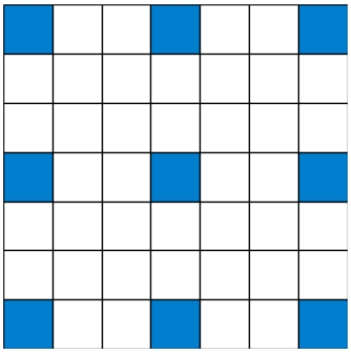
Problem 4.

Can you modify the the shape of the input shown in fig. 1, step by step, through a deconvolution process as shown in class? First, apply the conceptual transformations for the stride and output padding, then the shape modification after the kernel, and finally the shape modification from the padding parameter. Use a kernel size $K = 5$, a stride $S = 3$, an output padding $OP = 1$ and a padding $P = 1$.

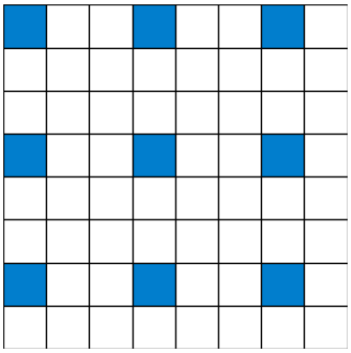
Solution: The solution is shown in fig. 2



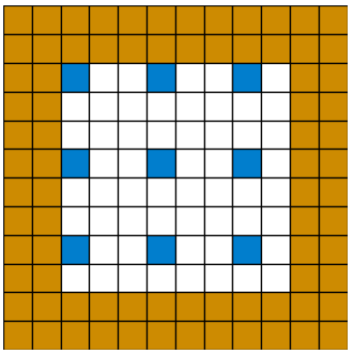
(a) Step 1: The original input image.



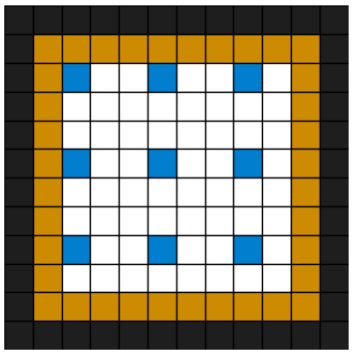
(b) Step 2: The image after applying the conceptual stride transformation, adding $S - 1 = 2$ zeros between elements.



(c) Step 3: Output padding is added now to the image, adding bottom and right borders of zeros.



(d) Step 4: After applying the kernel filter, the images is made larger by $K - 1 = 4$ features across each dimension.



(e) Step 5: The image after applying padding, removing borders of width $P = 1$.

Figure 2: Visualization of the transformation process through various steps of a tranposed convolution: input, stride, output padding, kernel application, and padding.