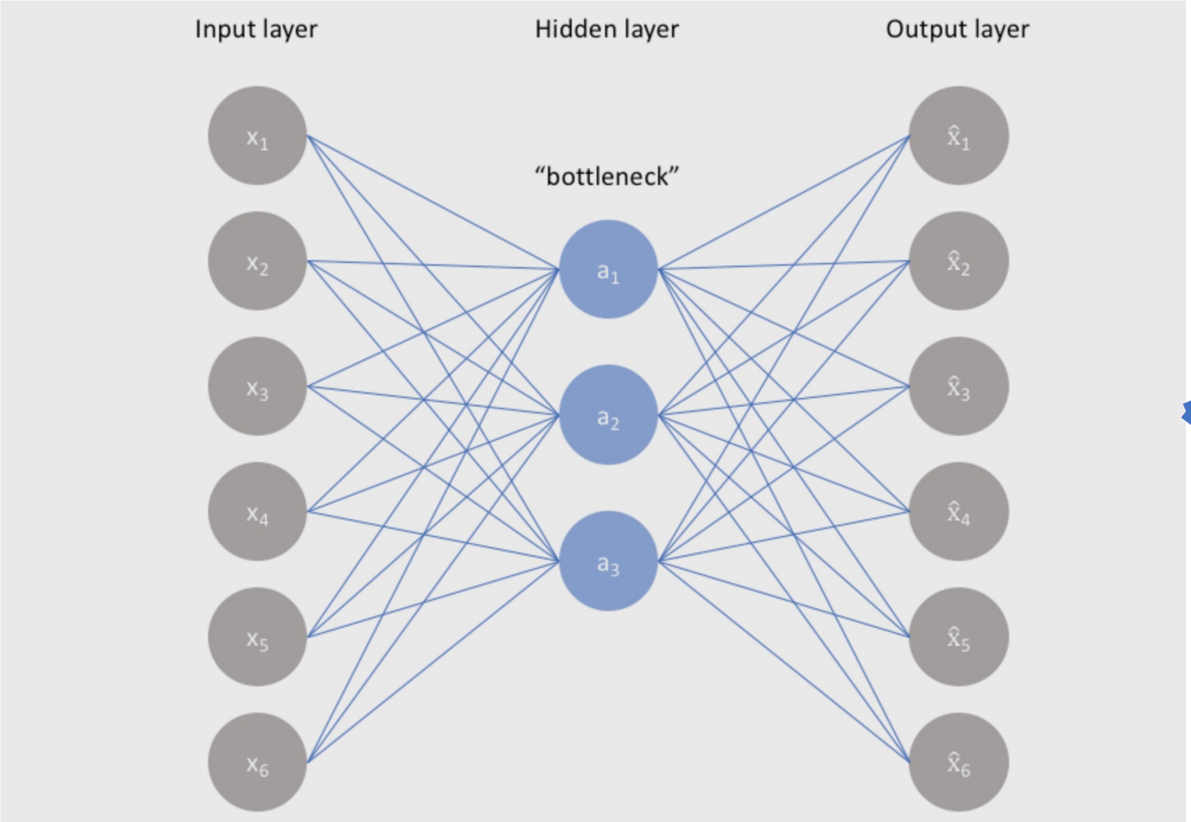
1. Autoencoders

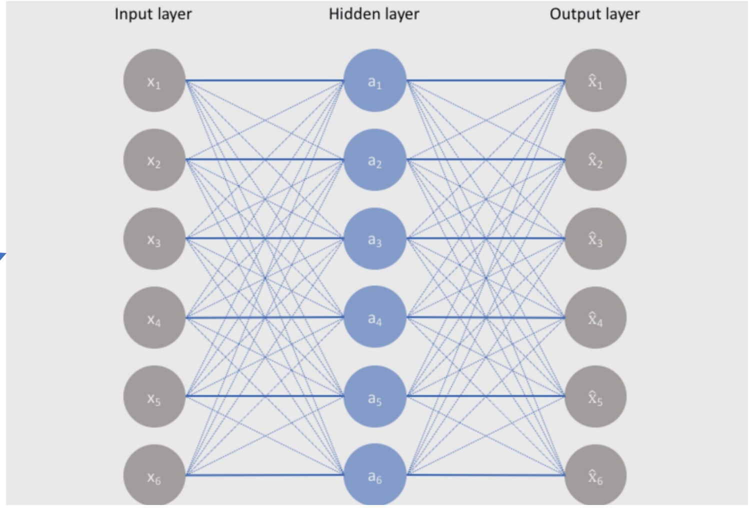


Reconstruction error to minimize:

$$\mathcal{L}(\theta, \phi) = \frac{1}{n}\sum_{i=1}^{n}(x^i - f_\theta(g_\phi(x^i)))^2$$

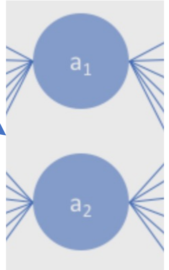
➡ **MSE**

Bottleneck essential:



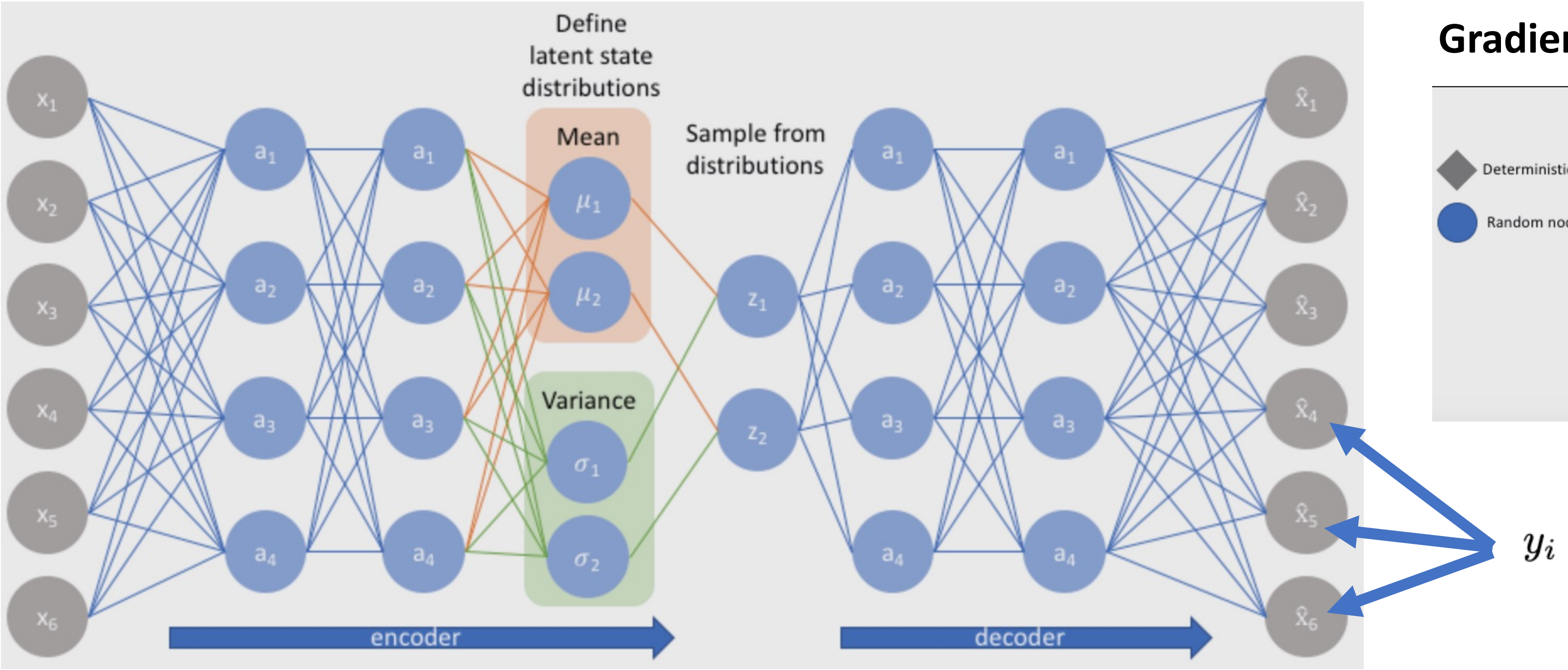
➡ Not learning anything

Activation functions

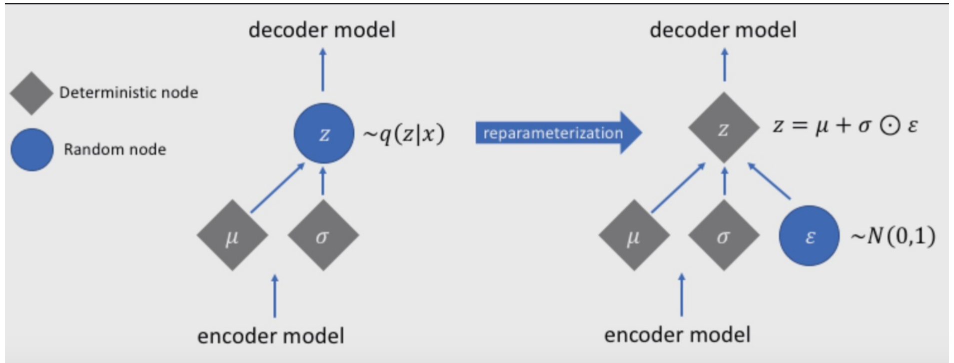


➡ Linear functions = similar to PCA

2. Variational autoencoders



Gradient computation



Reconstruction error to minimize:

$$\mathcal{L}(\theta,\phi) = \mathcal{R}(x^i, f_\theta(g_\phi(x^i))) + \beta D_{KL}(q_\phi(g_\phi(x^i)|x^i)||p_\theta(g_\phi(x^i))) \; = \mathcal{R}(x^i, y^i) + \beta D_{KL}(q_\phi(z^i)|x^i)||p_\theta(z^i))$$

Total cross-entropy

Entanglement

Kullback-Leibler divergence

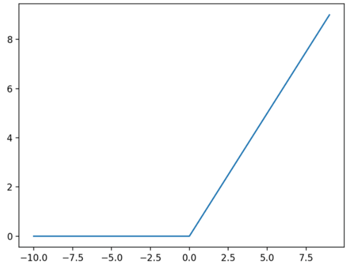
Cross-entropy loss:

$$\mathcal{L}_{CL} = -\sum_{i=1}^{n}\sum_{c=1}^{C} 1_{f_\theta(g_\phi(x_i))\in C_c} log(p(f_\theta(g_\phi(x_i)) \in C_c)) = -\sum_{i=1}^{n}\sum_{c=1}^{C} 1_{f_\theta(z_i)\in C_c} log(p(f_\theta(z_i) \in C_c))$$

$$= -\sum_{i=1}^{n}\sum_{c=1}^{C} 1_{y_i\in C_c} log(p(y_i \in C_c))$$

ReLu activation:

$$ReLu(x) = max(0, x)$$



KL divergence:

$$D_{KL}(q_\phi(z^k)|x^k)||p_\theta(z^k)) = \sum_{z_i\in z^k} q_\phi(z_i) log\left(\frac{q_\phi(z_i)}{p_\theta(z_i)}\right) = -\sum_{z_i\in z^k} q_\phi(z_i) log\left(\frac{p_\theta(z_i)}{q_\phi(z_i)}\right)$$

In our VAE model, the KL divergence is actually based on points and not whole distributions:

$$D_{KL}(q_\phi(z^k)|x^k)||p_\theta(z^k)) = \frac{1}{2}\sum_{i=1}^{N}(1 + log(\sigma_i^2) - (\mu_i^2) - (\sigma_i^2))$$

The parameters ϕ and θ of our encoder and decoder MLP are optimized jointly.

3 Example: Variational Auto-Encoder

In this section we'll give an example where we use a neural network for the probabilistic encoder $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ (the approximation to the posterior of the generative model $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})$) and where the parameters $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ are optimized jointly with the AEVB algorithm.

Let the prior over the latent variables be the centered isotropic multivariate Gaussian $p_{\boldsymbol{\theta}}(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$. Note that in this case, the prior lacks parameters. We let $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ be a multivariate Gaussian (in case of real-valued data) or Bernoulli (in case of binary data) whose distribution parameters are computed from $\mathbf{z}$ with a MLP (a fully-connected neural network with a single hidden layer, see appendix C). Note the true posterior $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ is in this case intractable. While there is much freedom in the form $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$, we'll assume the true (but intractable) posterior takes on a approximate Gaussian form with an approximately diagonal covariance. In this case, we can let the variational approximate posterior be a multivariate Gaussian with a diagonal covariance structure[2]:

$$\log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)}) = \log \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}^{(i)}, \boldsymbol{\sigma}^{2(i)}\mathbf{I}) \tag{9}$$

where the mean and s.d. of the approximate posterior, $\boldsymbol{\mu}^{(i)}$ and $\boldsymbol{\sigma}^{(i)}$, are outputs of the encoding MLP, i.e. nonlinear functions of datapoint $\mathbf{x}^{(i)}$ and the variational parameters $\boldsymbol{\phi}$ (see appendix C).

As explained in section 2.4, we sample from the posterior $\mathbf{z}^{(i,l)} \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})$ using $\mathbf{z}^{(i,l)} = g_{\boldsymbol{\phi}}(\mathbf{x}^{(i)}, \boldsymbol{\epsilon}^{(l)}) = \boldsymbol{\mu}^{(i)} + \boldsymbol{\sigma}^{(i)} \odot \boldsymbol{\epsilon}^{(l)}$ where $\boldsymbol{\epsilon}^{(l)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. With $\odot$ we signify an element-wise product. In this model both $p_{\boldsymbol{\theta}}(\mathbf{z})$ (the prior) and $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ are Gaussian; in this case, we can use the estimator of eq. (7) where the KL divergence can be computed and differentiated without estimation (see appendix B). The resulting estimator for this model and datapoint $\mathbf{x}^{(i)}$ is:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) \simeq \frac{1}{2} \sum_{j=1}^{J} \left(1 + \log((\sigma_j^{(i)})^2) - (\mu_j^{(i)})^2 - (\sigma_j^{(i)})^2 \right) + \frac{1}{L} \sum_{l=1}^{L} \log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}^{(i,l)})$$

$$\text{where} \quad \mathbf{z}^{(i,l)} = \boldsymbol{\mu}^{(i)} + \boldsymbol{\sigma}^{(i)} \odot \boldsymbol{\epsilon}^{(l)} \quad \text{and} \quad \boldsymbol{\epsilon}^{(l)} \sim \mathcal{N}(0, \mathbf{I}) \tag{10}$$

As explained above and in appendix C, the decoding term $\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}^{(i,l)})$ is a Bernoulli or Gaussian MLP, depending on the type of data we are modelling.

Auto-Encoding Variational Bayes, Diederik P Kingma, Max Welling, 2013

Literature review:

- **Auto-Encoding Variational Bayes**, Diederik P Kingma, Max Welling, 2013
- **Methods for the characterization of specificity in immune response to Sars-Cov-2**, Cheung Kaihung, M4R Project 2022
- **Deep generative models for T cell receptor protein sequences,** Kristian Davidsen et al., 2019
- **TITAN: T-cell receptor specificity prediction with bimodal attention networks**, Anna Weber, Jannis Born, María Rodriguez Martínez, 2021
- **TCR-BERT: learning the grammar of T-cell receptors for flexible antigen-xbinding analyses**, Kevin Wu et al., 2021
- **Quantitative Immunology for Physicists,** Grégoire Altan-Bonnet, Thierry Mora, Aleksandra M. Walczak, 2019
- **TCR meta-clonotypes for biomarker discovery with tcrdist3 enabled identification of public, HLA-restricted clusters of SARS-CoV-2 TCRs,** Koshlan Mayer-Blackwell et al., 2021
- **In silico proof of principle of machine learning-based antibody design at unconstrained scale,** Rahmad Akbar et al., 2022
- **A framework for highly multiplexed dextramer mapping and prediction of T cell receptor sequences to antigen specificity**, Wen Zhang et al., 2021
- **DECODE: a computational pipeline to discover T cell receptor binding rules**, Iliana Papadopoulou, An-Phi Nguyen, Anna Weber, María Rodríguez Martínez, 2022
- https://www.jeremyjordan.me/variational-autoencoders/
- https://medium.com/@j.zh/mathematics-behind-variational-autoencoders-c69297301957#:~:text=Variational%20Auto%2DEncoder(VAE),learning%20and%20dimensionality%20reduction%20etc.

List is not complete and updated though the writing of the report.