Understanding Deep Learning Errata

December 24, 2023

Much gratitude to everyone who has pointed out mistakes. If you find a problem not listed here, please contact me via github or by mailing me at udlbookmail@gmail.com.

First printing (Dec. 2023)

1.1 Errors

These are things that might genuinely confuse you.

- Figure 4.7b had the wrong calculated numbers in it (but pattern is same). Correct version is in figure 1.1 of this document.
- Section 7.5.1 The expectation (mean) $\mathbb{E}[f_i{}']$ of the intermediate values $f_i{}'$ is:
- Equation 15.9. First integrand should be with respect to $\mathbf{x}^*$. Correct version is:

$$\begin{aligned}
D_{JS}\Big[Pr(\mathbf{x}^*)\,||\; Pr(\mathbf{x})\Big] \\
&= \frac{1}{2}D_{KL}\left[Pr(\mathbf{x}^*)\middle|\middle|\frac{Pr(\mathbf{x}^*)+Pr(\mathbf{x})}{2}\right]+\frac{1}{2}D_{KL}\left[Pr(\mathbf{x})\middle|\middle|\frac{Pr(\mathbf{x}^*)+Pr(\mathbf{x})}{2}\right] \\
&= \frac{1}{2}\int \underbrace{Pr(\mathbf{x}^*)\log\left[\frac{2Pr(\mathbf{x}^*)}{Pr(\mathbf{x}^*)+Pr(\mathbf{x})}\right]d\mathbf{x}^*}_{\text{quality}}+\underbrace{\frac{1}{2}\int Pr(\mathbf{x})\log\left[\frac{2Pr(\mathbf{x})}{Pr(\mathbf{x}^*)+Pr(\mathbf{x})}\right]d\mathbf{x}}_{\text{coverage}}.
\end{aligned}$$

- Section 15.2.4 Consider distributions $Pr(x=i)$ and $q(x=j)$ defined over K bins. Assume there is a cost C_{ij} associated with moving one unit of mass from bin i in the first distribution to bin j in the second;
- Equation 15.14. Missing bracket and we don't need to use $\mathbf{x}^*$ notation here. Correct version is:

$$D_w\Big[Pr(\mathbf{x}), q(\mathbf{x})\Big] = \min_{\pi[\bullet,\bullet]}\left[\iint \pi(\mathbf{x}_1,\mathbf{x}_2)\cdot||\mathbf{x}_1-\mathbf{x}_2||d\mathbf{x}_1 d\mathbf{x}_2\right].$$

- Equation 15.15. Don't need to use $\mathbf{x}^*$ notation here, and second term on right hand side should have $q[\mathbf{x}]$ term not $Pr(\mathbf{x})$. Correct version is:

$$D_w\Big[Pr(\mathbf{x}), q(\mathbf{x})\Big] = \max_{\mathrm{f}[\mathbf{x}]}\left[\int Pr(\mathbf{x})\mathrm{f}[\mathbf{x}]d\mathbf{x} - \int q(\mathbf{x})\mathrm{f}[\mathbf{x}]d\mathbf{x}\right].$$

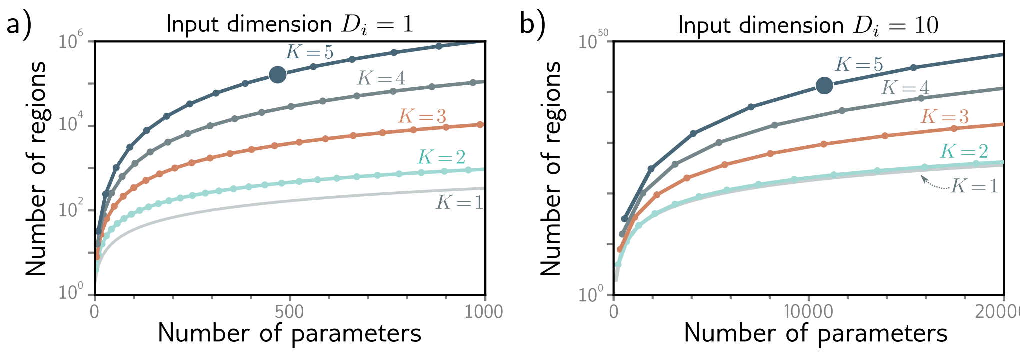


Figure 1.1 Corrected version of figure 4.7: The maximum number of linear regions for neural networks increases rapidly with the network depth. a) Network with $D_i = 1$ input. Each curve represents a fixed number of hidden layers K, as we vary the number of hidden units D per layer. For a fixed parameter budget (horizontal position), deeper networks produce more linear regions than shallower ones. A network with $K = 5$ layers and $D = 10$ hidden units per layer has 471 parameters (highlighted point) and can produce 161,051 regions. b) Network with $D_i = 10$ inputs. Each subsequent point along a curve represents ten hidden units. Here, a model with $K = 5$ layers and $D = 50$ hidden units per layer has 10,801 parameters (highlighted point) and can create more than 10^{40} linear regions.

- Equation 16.12 has a mistake in the second term. It should be:

$$\mathrm{f}[h_d, \boldsymbol{\phi}] = \left(\sum_{k=1}^{b-1} \phi_k\right) + (hK - b)\phi_b.$$

- Equation 17.34.

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{\phi}} \mathbb{E}_{Pr(x|\boldsymbol{\phi})}\big[\mathrm{f}[x]\big] &= \mathbb{E}_{Pr(x|\boldsymbol{\phi})}\left[\mathrm{f}[x]\frac{\partial}{\partial \boldsymbol{\phi}} \log\big[Pr(x|\boldsymbol{\phi})\big]\right] \\ &\approx \frac{1}{I}\sum_{i=1}^{I} \mathrm{f}[x_i]\frac{\partial}{\partial \boldsymbol{\phi}} \log\big[Pr(x_i|\boldsymbol{\phi})\big].\end{aligned}$$

- Figure 19.11 is wrong in that only the state-action values corresponding to the current state-action pair should be moderated. Correct version above.

- Appendix B.3.6. Consider a matrix $\mathbf{A} \in \mathbb{R}^{D_1 \times D_2}$. If the number of columns D_2 of the matrix is fewer than the number of rows D_1 (i.e., the matrix is "portrait"),

- Equation B.4. Square root sign should cover x. Correct version is:

$$x! \approx \sqrt{2\pi x}\left(\frac{x}{e}\right)^x.$$

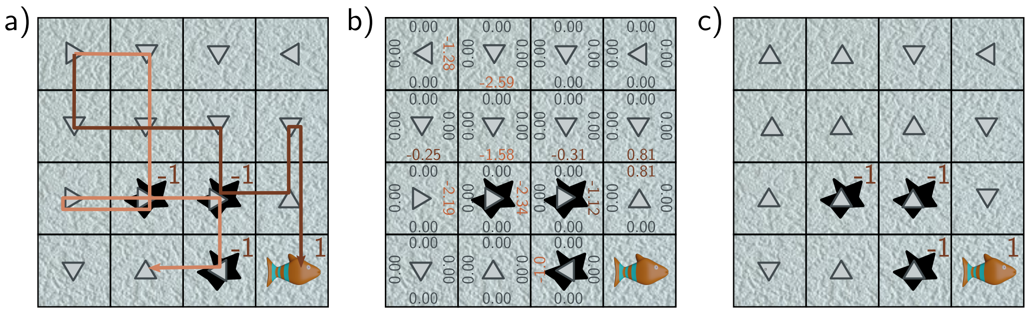


Figure 1.2 Corrected version of figure 19.11

- Equation C.20. Erroneous minus sign on covariance matrix. Correct version is:

$$\mathbf{x} = \boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\mathbf{z}.$$

1.1.1 Minor fixes

These are things that are wrong and need to be fixed, but that will probably not affect your understanding (e.g., math symbols that are in bold but should not be).

- Section 1.1: ...and what is meant by "training" a model.
- Figure 1.13: ~~Adapted from Pablok (2017).~~
- Figure 2.3 legend: Each combination of parameters $\boldsymbol{\phi} = [\phi_0, \phi_1]^T$.
- Section 2.3: 1D linear regression has the obvious drawback
- Figure 3.5 legend: The universal approximation theorem proves that, with enough hidden units, there exists a shallow neural network that can describe any given continuous function defined on a compact subset of $\mathbb{R}^{D_i}$ to arbitrary precision.
- Notes page 38 Most of these are attempts to avoid the dying ReLU problem while limiting the gradient for negative values.
- Figure 4.1 legend: The first network maps inputs $x \in [-1, 1]$ to outputs $y \in [-1, 1]$ using a function comprising three linear regions
- Equation 4.13 is missing a prime sign:

$$\begin{aligned}\mathbf{h} &= \mathbf{a}[\boldsymbol{\theta}_0 + \boldsymbol{\theta}x]\\ \mathbf{h}' &= \mathbf{a}[\boldsymbol{\psi}_0 + \boldsymbol{\Psi}\mathbf{h}]\\ y' &= \phi'_0 + \boldsymbol{\phi}'\mathbf{h}',\end{aligned}$$

- Equation 4.14: ϕ'_0 should not be bold.

$$y = \phi'_0 + \boldsymbol{\phi}'\mathbf{h}'$$

- Equation 4.17 is not technically wrong, but the product is unnecessary and it's unclear if the last term should be included in it (no). Better written as:

$$N_r = \left(\frac{D}{D_i} + 1\right)^{D_i(K-1)} \cdot \sum_{j=0}^{D_i} \binom{D}{j}.$$

- Equation 5.10. Second line is disambiguated by adding brackets:

$$\begin{aligned}\hat{\boldsymbol{\phi}} &= \underset{\boldsymbol{\phi}}{\operatorname{argmin}}\left[-\sum_{i=1}^{I}\log\left[\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left[-\frac{(y_i-\mathrm{f}[\mathbf{x}_i,\boldsymbol{\phi}])^2}{2\sigma^2}\right]\right]\right]\\ &= \underset{\boldsymbol{\phi}}{\operatorname{argmin}}\left[-\sum_{i=1}^{I}\left(\log\left[\frac{1}{\sqrt{2\pi\sigma^2}}\right]-\frac{(y_i-\mathrm{f}[\mathbf{x}_i,\boldsymbol{\phi}])^2}{2\sigma^2}\right)\right]\\ &= \underset{\boldsymbol{\phi}}{\operatorname{argmin}}\left[-\sum_{i=1}^{I}-\frac{(y_i-\mathrm{f}[\mathbf{x}_i,\boldsymbol{\phi}])^2}{2\sigma^2}\right]\\ &= \underset{\boldsymbol{\phi}}{\operatorname{argmin}}\left[\sum_{i=1}^{I}(y_i-\mathrm{f}[\mathbf{x}_i,\boldsymbol{\phi}])^2\right],\end{aligned}$$

- Equation 5.15. Disambiguated by adding brackets:

$$\hat{\boldsymbol{\phi}} = \underset{\boldsymbol{\phi}}{\operatorname{argmin}}\left[-\sum_{i=1}^{I}\left(\log\left[\frac{1}{\sqrt{2\pi\mathrm{f}_2[\mathbf{x}_i,\boldsymbol{\phi}]^2}}\right]-\frac{(y_i-\mathrm{f}_1[\mathbf{x}_i,\boldsymbol{\phi}])^2}{2\mathrm{f}_2[\mathbf{x}_i,\boldsymbol{\phi}]^2}\right)\right].$$

- Section 5.6 Removed i index from this paragraph for consistency. Independence implies that we treat the probability $Pr(\mathbf{y}|\mathbf{f}[\mathbf{x},\boldsymbol{\phi}])$ as a product of univariate terms for each element $y_d \in \mathbf{y}$:

$$Pr(\mathbf{y}|\mathbf{f}[\mathbf{x},\boldsymbol{\phi}]) = \prod_d Pr(y_d|\mathbf{f}_d[\mathbf{x},\boldsymbol{\phi}]),$$

  where $\mathbf{f}_d[\mathbf{x},\boldsymbol{\phi}]$ is the d^{th} set of network outputs, which describe the parameters of the distribution over y_d. For example, to predict multiple continuous variables $y_d \in \mathbb{R}$, we use a normal distribution for each y_d, and the network outputs $\mathrm{f}_d[\mathbf{x},\boldsymbol{\phi}]$ predict the means of these distributions. To predict multiple discrete variables $y_d \in \{1,2,\ldots,K\}$, we use a categorical distribution for each y_d. Here, each set of network outputs $\mathbf{f}_d[\mathbf{x},\boldsymbol{\phi}]$ predicts the K values that contribute to the categorical distribution for y_d.

- Problem 5.8. Construct a loss function for making multivariate predictions $\mathbf{y} \in \mathbb{R}^{D_i}$ based on independent normal distributions...

- Notes page 94. However, this is strange since SGD is a special case of Adam (when $\beta = \gamma = 0$)

- Section 7.4. Similarly, the derivative for the weights matrix $\boldsymbol{\Omega}_k$, is given by

- Section 8.4.1 When the number of parameters is very close to the number of training data examples (figure 8.11b)

- Figure 9.11 legend: a-c) Two sets of parameters (cyan and gray curves) sampled from the posterior

- Section 10.2.1 Not wrong, but could be disambiguated: The size of the region over which inputs are combined is termed the *kernel size*.
- Section 10.2.3 The number of zeros we intersperse between the weights determines the dilation rate.
- Section 10.2.4 With kernel size three, stride one, and dilation rate one.
- Figure 10.3. The dilation rates are wrong by one, so should be 1,1,1, and 2 in panels a,b,c,d, respectively.
- Section 10.5.3 The first part of the network is a smaller version of VGG (figure 10.17) that contains thirteen rather than sixteen convolutional layers.
- Section 10.6 The weights and the bias are the same at every spatial position, so there are far fewer parameters than in a fully connected network, and the number of parameters doesn't increase with the input image size.
- Problem 10.2 Equation 10.3 defines 1D convolution with a kernel size of three, stride of one, and dilation one.
- Problem 10.3 Write out the equation for the 1D dilated convolution with a kernel size of three and a dilation rate of two.
- Problem 10.4 Write out the equation for a 1D convolution with kernel size of seven, a dilation rate of three, and a stride of three.
- Problem 10.9 A network consists of three 1D convolutional layers. At each layer, a zero-padded convolution with kernel size three, stride one, and dilation one is applied.
- Problem 10.10 A network consists of three 1D convolutional layers. At each layer, a zero-padded convolution with kernel size seven, stride one, and dilation one is applied.
- Problem 10.11 Consider a convolutional network with 1D input $\mathbf{x}$. The first hidden layer $\mathbf{H}_1$ is computed using a convolution with kernel size five, stride two, and a dilation rate of one. The second hidden layer $\mathbf{H}_2$ is computed using a convolution with kernel size three, stride one, and a dilation rate of one. The third hidden layer $\mathbf{H}_3$ is computed using a convolution with kernel size five, stride one, and a dilation rate of two. What are the receptive field sizes at each hidden layer?
- Legend to figure 11.15. Computational graph for batch normalization (see problem 11.5).
- Section 12.2.2: Not a mistake, but this is clearer: where $\boldsymbol{\beta}_v \in \mathbb{R}^D$ and $\boldsymbol{\Omega}_v \in \mathbb{R}^{D\times D}$ represent biases and weights, respectively.
- Section 12.3.3 to make self-attention work well

- Section 12.4 a series of these transformer layers ...
- Section 12.5 The previous section described the transformer layer... a series of transformer layers...
- Figure 12.8 legend: The transformer → Transformer layer...The transformer layer consists
- Figure 12.8 has some minor mistakes in the calculation. The corrected version is shown at the end of this document.
- Figure 12.8 legend. At each iteration, the sub-word tokenizer looks for the most commonly occurring adjacent pair of tokens
- Section 12.5.3 a series of K transformer layers
- Section 12.6 through 24 transformer layers
- Section 12.6 in the fully connected networks ~~in the transformer~~ is 4096
- Figure 12.10 a series of transformer layers
- Section 12.7.2 the transformer layers use masked...
- Figure 12.12 are passed through a series of transformer layers... and those of tokens earlier
- Section 12.7.4 There are 96 transformer layers
- Section 12.7 comprises a series of transformer layers
- Section 12.8 Originally, these
- Section 12.8 a series of transformer layers... a series of transformer layers
- Equation 15.6. Minor problems with brackets in this equation. Should be:

$$
\begin{aligned}
L[\boldsymbol{\phi}] &= \frac{1}{J}\sum_{j=1}^{J}\left(\log\left[1-\text{sig}[\text{f}[\mathbf{x}_j^*,\boldsymbol{\phi}]]\right]\right)+\frac{1}{I}\sum_{i=1}^{I}\left(\log\left[\text{sig}[\text{f}[\mathbf{x}_i,\boldsymbol{\phi}]]\right]\right)\\
&\approx \mathbb{E}_{\mathbf{x}^*}\left[\log\left[1-\text{sig}[\text{f}[\mathbf{x}^*,\boldsymbol{\phi}]]\right]\right]+\mathbb{E}_{\mathbf{x}}\left[\log\left[\text{sig}[\text{f}[\mathbf{x},\boldsymbol{\phi}]]\right]\right]\\
&= \int Pr(\mathbf{x}^*)\log\left[1-\text{sig}[\text{f}[\mathbf{x}^*,\boldsymbol{\phi}]]\right]d\mathbf{x}^*+\int Pr(\mathbf{x})\log\left[\text{sig}[\text{f}[\mathbf{x},\boldsymbol{\phi}]]\right]d\mathbf{x}.
\end{aligned}
$$

- Equation 16.2 (last line). For some reason, this didn't print properly, although it looks fine in my original pdf. Should be:

$$\begin{aligned}
\hat{\boldsymbol{\phi}} &= \underset{\boldsymbol{\phi}}{\operatorname{argmax}}\left[\prod_{i=1}^{I} Pr(x_i|\boldsymbol{\phi})\right] \\
&= \underset{\boldsymbol{\phi}}{\operatorname{argmin}}\left[\sum_{i=1}^{I} -\log\Big[Pr(x_i|\boldsymbol{\phi})\Big]\right] \\
&= \underset{\boldsymbol{\phi}}{\operatorname{argmin}}\left[\sum_{i=1}^{I} \log\left[\left|\frac{\partial \mathrm{f}[z_i,\boldsymbol{\phi}]}{\partial z_i}\right|\right] - \log\Big[Pr(z_i)\Big]\right],
\end{aligned}$$

- Equation 16.25. $\boldsymbol{\phi}$ should change to $\hat{\boldsymbol{\phi}}$ on left hand side. Correct version is:

$$\hat{\boldsymbol{\phi}} = \underset{\boldsymbol{\phi}}{\operatorname{argmin}}\left[\mathrm{KL}\left[\sum_{i=1}^{I}\delta\left[\mathbf{x}-\mathrm{f}[\mathbf{z}_i,\boldsymbol{\phi}]\right]\Big|\Big|q(\mathbf{x})\right]\right].$$

- Equation 16.26. $\boldsymbol{\phi}$ should change to $\hat{\boldsymbol{\phi}}$ on left hand side. Correct version is:

$$\hat{\boldsymbol{\phi}} = \underset{\boldsymbol{\phi}}{\operatorname{argmin}}\left[\mathrm{KL}\left[\frac{1}{I}\sum_{i=1}^{I}\delta[\mathbf{x}-\mathbf{x}_i]\Big|\Big|Pr(\mathbf{x}_i,\boldsymbol{\phi})\right]\right].$$

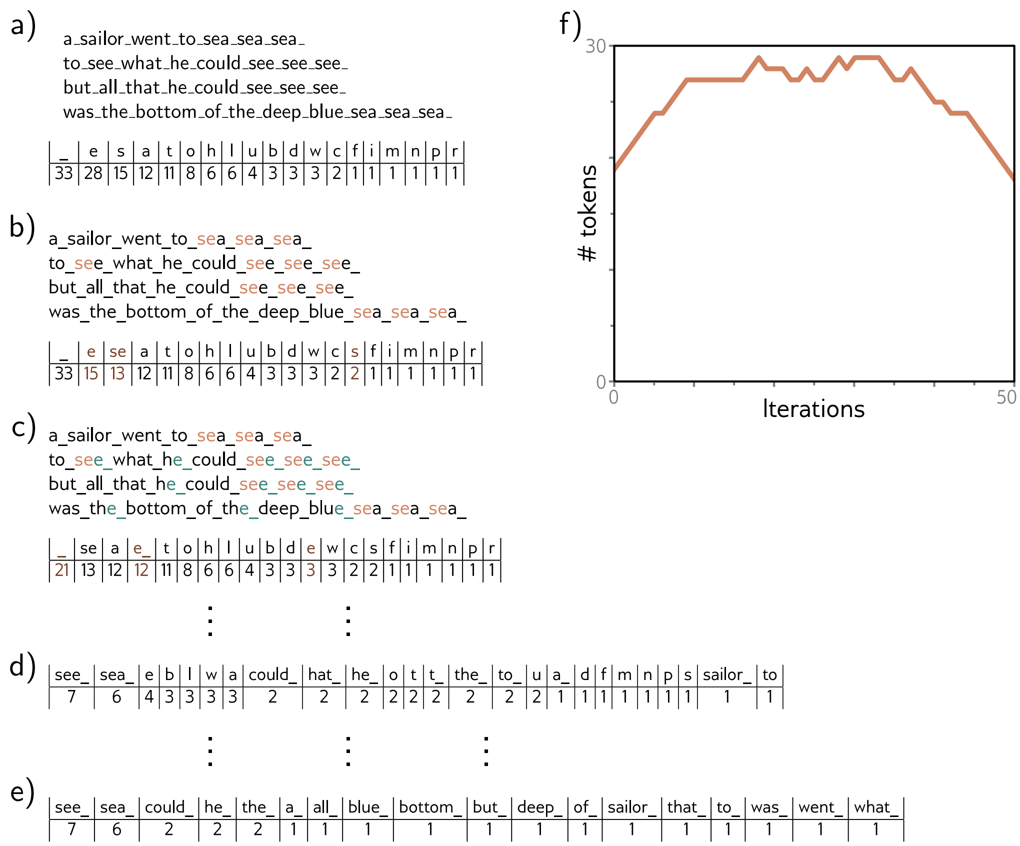
- Equation 18.24 has a minor formatting mistake. Better written as:

$$\begin{aligned}
&\log\left[\frac{Pr(\mathbf{x},\mathbf{z}_{1\ldots T}|\boldsymbol{\phi}_{1\ldots T})}{q(\mathbf{z}_{1\ldots T}|\mathbf{x})}\right] \\
&\quad= \log\left[\frac{Pr(\mathbf{x}|\mathbf{z}_1,\boldsymbol{\phi}_1)}{q(\mathbf{z}_1|\mathbf{x})}\right] + \log\left[\frac{\prod_{t=2}^{T} Pr(\mathbf{z}_{t-1}|\mathbf{z}_t,\boldsymbol{\phi}_t)\cdot q(\mathbf{z}_{t-1}|\mathbf{x})}{\prod_{t=2}^{T} q(\mathbf{z}_{t-1}|\mathbf{z}_t,\mathbf{x})\cdot q(\mathbf{z}_t|\mathbf{x})}\right] + \log\Big[Pr(\mathbf{z}_T)\Big] \\
&\quad= \log\left[Pr(\mathbf{x}|\mathbf{z}_1,\boldsymbol{\phi}_1)\right] + \log\left[\frac{\prod_{t=2}^{T} Pr(\mathbf{z}_{t-1}|\mathbf{z}_t,\boldsymbol{\phi}_t)}{\prod_{t=2}^{T} q(\mathbf{z}_{t-1}|\mathbf{z}_t,\mathbf{x})}\right] + \log\left[\frac{Pr(\mathbf{z}_T)}{q(\mathbf{z}_T|\mathbf{x})}\right] \\
&\quad\approx \log\left[Pr(\mathbf{x}|\mathbf{z}_1,\boldsymbol{\phi}_1)\right] + \sum_{t=2}^{T}\log\left[\frac{Pr(\mathbf{z}_{t-1}|\mathbf{z}_t,\boldsymbol{\phi}_t)}{q(\mathbf{z}_{t-1}|\mathbf{z}_t,\mathbf{x})}\right],
\end{aligned}$$

- Section 20.5.1 In general, the smaller the model, the larger the proportion of weights that can...

- Section 20.22 Another possible explanation for the ease with which models are trained is that some regularization methods like L2 regularization (weight decay) make the loss surface flatter and more convex.

- Appendix A. The notation $\{0, 1, 2, \ldots\}$ denotes the set of non-negative integers.



Corrected version of figure 12.8

- Appendix A ...*big-O* notation, which represents an upper bound...
- Appendix A. $\mathrm{f}[n] < c\cdot\mathrm{g}[n]$ for all $n > n_0$
- Appendix C.5.4 Accent in wrong place: The Fréchet and Wasserstein distances...
- Equation C.32.

$$
\begin{aligned}
D_{KL}\Big[\mathrm{Norm}[\boldsymbol{\mu}_1,\boldsymbol{\Sigma}_1]\Big|\Big|\mathrm{Norm}[\boldsymbol{\mu}_2,\boldsymbol{\Sigma}_2]\Big] \quad &= \\
&\frac{1}{2}\left(\log\left[\frac{|\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|}\right] - D + \mathrm{tr}\left[\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_1\right] + (\boldsymbol{\mu}_2-\boldsymbol{\mu}_1)\boldsymbol{\Sigma}_2^{-1}(\boldsymbol{\mu}_2-\boldsymbol{\mu}_1)\right]\right).
\end{aligned}
$$