Understanding Deep Learning Errata

July 2, 2024

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.

Instructions

To find which errata are relevant to your version of the book, first consult the copyright page at the start of the book just before the dedication. The printing will be stated (e.g., "Second printing") just before the line that says "Library of Congress...". If it doesn't specify the printing here, then you have the first printing.

This document is organized by printing. If you have the first printing of the book, all errata are relevant to you. If you have the second printing, then only the errata in the sections for second and third printings are relevant. If you have the third printing, then only the errata for the third printing is relevant and so on.

Chapter 2

Fourth printing (Jun 2024?)

Errors

These are things that might genuinely confuse you:

- Problem 15.1: What will the loss be in equation 15.9 when $Pr(\mathbf{x}^*) = Pr(\mathbf{x})$?
- Equation 16.22:

$$\begin{aligned}\log\left[\left|\mathbf{I}+\frac{\partial \mathbf{f}[\mathbf{h},\boldsymbol{\phi}]}{\partial \mathbf{h}}\right|\right] &= \text{trace}\left[\log\left[\mathbf{I}+\frac{\partial \mathbf{f}[\mathbf{h},\boldsymbol{\phi}]}{\partial \mathbf{h}}\right]\right]\\ &= \sum_{k=1}^{\infty}\frac{(-1)^{k-1}}{k}\text{trace}\left[\frac{\partial \mathbf{f}[\mathbf{h},\boldsymbol{\phi}]}{\partial \mathbf{h}}\right]^{k},\end{aligned}$$

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):

- Chapter 1 introduction: Hence, this chapter also contains a brief primer on AI ethics.
- Figure 4.6 legend: The weights are stored in matrices $\boldsymbol{\Omega}_k$ that ~~pre-~~multiply the activations from the preceding layer.
- Section 5.2: To perform inference for a new test example $\mathbf{x}$, return either the full distribution $Pr(\mathbf{y}|\mathbf{f}[\mathbf{x},\hat{\boldsymbol{\phi}}])$ or the value where this distribution is maximized.

- Section 5.3.1: We see that the least squares loss function follows naturally from the assumptions that the predictions are (i) independent and (ii) drawn from a normal distribution with mean $\mu = \mathrm{f}[\mathbf{x}_i, \boldsymbol{\phi}]$.
- Section 5.3.1: We removed the denominator between the third and fourth lines, as this is just a constant positive scaling factor that does not affect the position of the minimum. (other minor changes to preserve formatting)
- Section 5.3.3 However, there is nothing to stop us from treating σ^2 as a learned parameter ~~of the model~~and minimizing equation 5.9 with...
- Figure 5.12: Model distribution (a normal distribution with parameters $\boldsymbol{\theta} = \{\mu, \sigma^2\}$).
- Section 7.4 Now let's consider how the loss changes when the pre-activations $\mathbf{f}_0, \mathbf{f}_1, \mathbf{f}_2$ change.
- Equations 7.29 and 7.30 + nearby text. Added subscript to variance so $\sigma^2_{f'} \to \sigma^2_{f'_i}$.
- Section 8.1 This is better than the chance error rate of 90% ~~error rate~~ but far worse than for the training set;
- Section 9.3: We've seen that ~~adding~~ explicit regularization ~~terms~~ encourages the training algorithm to find a good solution by adding extra terms to the loss function.
- Chapter 11 Notes p202. A second ReLU function was applied after this block and was added back to the main representation. This architecture was termed *ResNet v1*.
- Section 15.2.6: Changed to stop nasty line wrap-over: One way to achieve this is to clip the discriminator weights to a small range (e.g., ± 0.01)
- Problem 15.2: Write an equation relating the loss L in equation 15.8 to the Jensen-Shannon distance $D_{JS}\left[Pr(\mathbf{x}^*) \,||\, Pr(\mathbf{x})\right]$ in equation 15.9.
- Section 16.3.3: If the function $\mathbf{g}[\bullet, \boldsymbol{\phi}]$ is an elementwise flow, the Jacobian will be lower triangular ~~diagonal~~with the identity matrix in the top-left quadrant and the derivatives of the elementwise transformation in the bottom-right.
- Figure 19.14: Input should be tagged as having size $84 \times 84 \times 4$.
- Section 19.4.2: ...leads to a systematic bias in the estimated action values $q[s_t, a_t]$.
- Equation 19.30: Changed for compatibility with earlier definition of reward:

$$r[\boldsymbol{\tau}_i] = \sum_{t=1}^{T} r_{i,t+1} = \sum_{k=1}^{t} r_{i,k+1} + \sum_{k=t}^{T} r_{i,k+1},$$

- Equation 19.31: Changed for compatibility with earlier definition of reward:

$$\boldsymbol{\theta} \;\leftarrow\; \boldsymbol{\theta} + \alpha \cdot \frac{1}{I}\sum_{i=1}^{I}\sum_{t=1}^{T}\frac{\partial \log\left[\pi[a_{it}|\mathbf{s}_{it}, \boldsymbol{\theta}]\right]}{\partial \boldsymbol{\theta}} \sum_{k=t}^{T} r_{i,k+1}.$$

- Equation 19.32: Changed for compatibility with earlier definition of reward:

$$r[\boldsymbol{\tau}_{it}] = \sum_{k=t+1}^{T} \gamma^{k-t-1} r_{i,k+1},$$

- Equation 19.39: Changed for compatibility with earlier definition of reward:

$$L[\boldsymbol{\phi}] = \sum_{i=1}^{I}\sum_{t=1}^{T}\left(v[\mathbf{s}_{it},\boldsymbol{\phi}] - \sum_{j=t}^{T} r_{i,j+1}\right)^2.$$

- Equation 19.40: Changed for compatibility with earlier definition of reward:

$$r[\boldsymbol{\tau}_{it}] \approx r_{i,t+1} + \gamma\cdot v[\mathbf{s}_{i,t+1},\boldsymbol{\phi}].$$

- Equation 19.41: Changed for compatibility with earlier definition of reward:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha\cdot\frac{1}{I}\sum_{i=1}^{I}\sum_{t=1}^{T}\frac{\partial \log\bigl[Pr(a_{it}|\mathbf{s}_{it},\boldsymbol{\theta})\bigr]}{\partial\boldsymbol{\theta}}\Bigl(r_{i,t+1} + \gamma\cdot v[\mathbf{s}_{i,t+1},\boldsymbol{\phi}] - v[\mathbf{s}_{i,t},\boldsymbol{\phi}]\Bigr). \tag{2.1}$$

- Equation 19.42: Changed for compatibility with earlier definition of reward:

$$L[\boldsymbol{\phi}] = \sum_{i=1}^{I}\sum_{t=1}^{T}\left(r_{i,t+1} + \gamma\cdot v[\mathbf{s}_{i,t+1},\boldsymbol{\phi}] - v[\mathbf{s}_{i,t},\boldsymbol{\phi}]\right)^2. \tag{2.2}$$

Second/third printings (Mar. 2024)

Errors

These are things that might genuinely confuse you:

- Equation 6.12

$$
\begin{aligned}
\mathbf{m}_{t+1} &\leftarrow \beta \cdot \mathbf{m}_t + (1-\beta) \sum_{i \in \mathcal{B}_t} \frac{\partial \ell_i[\boldsymbol{\phi}_t - \alpha\beta \cdot \mathbf{m}_t]}{\partial \boldsymbol{\phi}} \\
\boldsymbol{\phi}_{t+1} &\leftarrow \boldsymbol{\phi}_t - \alpha \cdot \mathbf{m}_{t+1},
\end{aligned} \tag{2.3}
$$

- Equation 6.18:

$$
\begin{aligned}
\mathbf{m}_{t+1} &\leftarrow \beta \cdot \mathbf{m}_t + (1-\beta) \sum_{i \in \mathcal{B}_t} \frac{\partial \ell_i[\boldsymbol{\phi}_t]}{\partial \boldsymbol{\phi}} \\
\mathbf{v}_{t+1} &\leftarrow \gamma \cdot \mathbf{v}_t + (1-\gamma) \left(\sum_{i \in \mathcal{B}_t} \frac{\partial \ell_i[\boldsymbol{\phi}_t]}{\partial \boldsymbol{\phi}} \right)^2,
\end{aligned} \tag{2.4}
$$

- Equation 15.6

$$
\begin{aligned}
L[\boldsymbol{\phi}] &= -\frac{1}{J} \sum_{j=1}^{J} \left(\log\left[1 - \text{sig}[\text{f}[\mathbf{x}_j^*, \boldsymbol{\phi}]]\right] \right) - \frac{1}{I} \sum_{i=1}^{I} \left(\log\left[\text{sig}[\text{f}[\mathbf{x}_i, \boldsymbol{\phi}]]\right] \right) \qquad (2.5) \\
&\approx -\mathbb{E}_{\mathbf{x}^*}\left[\log\left[1 - \text{sig}[\text{f}[\mathbf{x}^*, \boldsymbol{\phi}]]\right]\right] - \mathbb{E}_{\mathbf{x}}\left[\log\left[\text{sig}[\text{f}[\mathbf{x}, \boldsymbol{\phi}]]\right]\right] \\
&= -\int Pr(\mathbf{x}^*) \log\left[1 - \text{sig}[\text{f}[\mathbf{x}^*, \boldsymbol{\phi}]]\right] d\mathbf{x}^* - \int Pr(\mathbf{x}) \log\left[\text{sig}[\text{f}[\mathbf{x}, \boldsymbol{\phi}]]\right] d\mathbf{x},
\end{aligned}
$$

- Equation 15.8

$$
\begin{aligned}
L[\boldsymbol{\phi}] &= -\int Pr(\mathbf{x}^*)\log\Big[1-\text{sig}[\text{f}[\mathbf{x}^*,\boldsymbol{\phi}]]\Big]d\mathbf{x}^* - \int Pr(\mathbf{x})\log\Big[\text{sig}[\text{f}[\mathbf{x},\boldsymbol{\phi}]]\Big]d\mathbf{x} \qquad (2.6)\\
&= -\int Pr(\mathbf{x}^*)\log\left[1-\frac{Pr(\mathbf{x})}{Pr(\mathbf{x}^*)+Pr(\mathbf{x})}\right]d\mathbf{x}^* - \int Pr(\mathbf{x})\log\left[\frac{Pr(\mathbf{x})}{Pr(\mathbf{x}^*)+Pr(\mathbf{x})}\right]d\mathbf{x}\\
&= -\int Pr(\mathbf{x}^*)\log\left[\frac{Pr(\mathbf{x}^*)}{Pr(\mathbf{x}^*)+Pr(\mathbf{x})}\right]d\mathbf{x}^* - \int Pr(\mathbf{x})\log\left[\frac{Pr(\mathbf{x})}{Pr(\mathbf{x}^*)+Pr(\mathbf{x})}\right]d\mathbf{x}.
\end{aligned}
$$

- Equation 19.40:

$$r[\boldsymbol{\tau}_{it}] \approx r_{it} + \gamma \cdot v[\mathbf{s}_{i,t+1},\boldsymbol{\phi}].$$

- Section B.3.6: Definition of nullspace was ambiguous/wrong. Last line of this section changed to: Conversely, for a landscape matrix $\mathbf{A}$, the part of the input space that maps to zero (i.e., those $\mathbf{x}$ where $\mathbf{Ax}=\mathbf{0}$) is termed the *nullspace* of the matrix.

- Section C.5.3: The definition of the Fréchet distance was incorrect. Changed to:

$$D_{Fr}\Big[p(x)\big|\big|q(y)\Big] = \sqrt{\min_{\pi(x,y)}\left[\iint \pi(x,y)|x-y|^2 dxdy\right]}, \qquad (2.7)$$

where $\pi(x,y)$ represents the set of joint distributions that are compatible with the marginal distributions $p(x)$ and $q(y)$. The Fréchet distance can also be formulated as a measure of the maximum distance between the cumulative probability curves.

- Equation C.32:

$$
\begin{aligned}
&D_{KL}\Big[\text{Norm}[\boldsymbol{\mu}_1,\boldsymbol{\Sigma}_1]\Big|\Big|\text{Norm}[\boldsymbol{\mu}_2,\boldsymbol{\Sigma}_2]\Big] \quad = \\
&\qquad \frac{1}{2}\left(\log\left[\frac{|\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|}\right] - D + \text{tr}\left[\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_1\right] + (\boldsymbol{\mu}_2-\boldsymbol{\mu}_1)\boldsymbol{\Sigma}_2^{-1}(\boldsymbol{\mu}_2-\boldsymbol{\mu}_1)\cancel{\big]}\right). \qquad (2.8)
\end{aligned}
$$

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 1.1.1: In contrast, the model in figure 1.2b takes the chemical structure of a molecule as an input and predicts both the freezing and boiling points.

- Figure 1.2 legend: This *multivariate regression* model takes the structure of a chemical molecule and predicts its freezing and boiling points.
- Section 1.1.2: Finally, consider the input for the model that predicts the freezing and boiling points of the molecule. A molecule may contain varying numbers of atoms that can
- Multiple places in Chapters 2-9. Loss functions $L[\boldsymbol{\phi}]$ sometimes written as $\mathrm{L}[\boldsymbol{\phi}]$. Have now all been converted to italic for consistency.
- Section 4.5.5 Added missing definition of over-parameterization and other minor changes for typesetting consistence: It may be that over-parameterized deep models (i.e., those with more parameters than training examples) have a large family...
- Chapter 4 Notes, page 52. Montúfar
- Problem 4.7: Choose values for the parameters $\boldsymbol{\phi} = \{\phi_0, \phi_1, \phi_2, \phi_3, \theta_{10}, \theta_{11}, \theta_{20}, \theta_{21}, \theta_{30}, \theta_{31}\}$ for the shallow neural network in equation 3.1 (with ReLU activation functions) that will define an identity function over a finite range $x \in [a, b]$.
- Equation 4.11: How many parameters does each network have? How many linear regions can each network make (see equation 4.17)?
- Problem 5.1 Reworded to be more precise with limits: Show that the logistic sigmoid function $\mathrm{sig}[z]$ becomes 0 as $z \to -\infty$, is 0.5 when $z = 0$, and becomes 1 when $z \to \infty$.
- Problem 5.3: The term $\mathrm{Bessel}_0[\kappa]$ is a modified Bessel function of the first kind of order 0.
- Problem 5.8: Construct a loss function for making multivariate predictions $\mathbf{y} \in \mathbb{R}^{D_o}$ based on...
- Figure 7.4 legend: We work backward from the end of the function computing the derivatives $\partial \ell_i / \partial f_k$ and $\partial \ell_i / \partial h_k$ of the loss with respect to the intermediate quantities.
- Section 7.3 Finally, we consider how the loss ℓ_i changes when we change the parameters $\{\beta_k\}$ and $\{\omega_k\}$.
- Figure 7.5 legend: Finally, we compute the derivatives $\partial \ell_i / \partial \beta_k$ and $\partial \ell_i / \partial \omega_k$
- Problem 7.13 For the same function as in problem 7.12, compute the derivative...
- Equation 8.2:

$$\begin{aligned} L[x] &= \big(\mathrm{f}[x, \boldsymbol{\phi}] - y[x]\big)^2 \\ &= \Big(\big(\mathrm{f}[x, \boldsymbol{\phi}] - \mu[x]\big) + \big(\mu[x] - y[x]\big)\Big)^2 \\ &= \big(\mathrm{f}[x, \boldsymbol{\phi}] - \mu[x]\big)^2 + 2\big(\mathrm{f}[x, \boldsymbol{\phi}]\cancel{)} - \mu[x]\big)\big(\mu[x] - y[x]\big) + \big(\mu[x] - y[x]\big)^2, \end{aligned} \tag{2.9}$$

- Section 8.4.2 There would be 10^{40} bins in total, constrained by only 10^{4} examples.
- Equation 9.6 LHS should be total derivative, not partial derivative:

$$\frac{d\boldsymbol{\phi}}{dt} = -\frac{\partial L}{\partial \boldsymbol{\phi}}. \tag{2.10}$$

- Figure 9.4 Added to legend for panel (a) Blue point represents global minimum. Added to legend for panel (d) Blue point represents global minimum which may now be in a different place from panel (a).
- Figure 9.9 legend Here we are using full-batch gradient descent, and the model (from figure 8.4) fits the data as well as possible, so further training won't remove the kink
- Figure 9.9 legend Consider what happens if we remove the eighth hidden unit
- Figure 9.11 legend: a-c) Two sets of parameters (cyan and gray curves) sampled from the posterior
- Figure 9.11 legend When the prior variance $\sigma^2_{\boldsymbol{\phi}}$ is small
- Equation 9.15 Should be total derivative not partial:

$$\frac{d\boldsymbol{\phi}}{dt} = \mathbf{g}[\boldsymbol{\phi}]. \tag{2.11}$$

- Equation 9.16 Should be total derivative not partial:

$$\frac{d\boldsymbol{\phi}}{dt} \approx \mathbf{g}[\boldsymbol{\phi}] + \alpha\mathbf{g}_1[\boldsymbol{\phi}] + \ldots, \tag{2.12}$$

- Equation 9.17 Should be total derivative not partial:

$$\begin{aligned}
\boldsymbol{\phi}[\alpha] &\approx \boldsymbol{\phi} + \alpha\frac{d\boldsymbol{\phi}}{dt} + \frac{\alpha^2}{2}\frac{d^2\boldsymbol{\phi}}{dt^2}\bigg|_{\boldsymbol{\phi}=\boldsymbol{\phi}_0} \\
&\approx \boldsymbol{\phi} + \alpha\left(\mathbf{g}[\boldsymbol{\phi}] + \alpha\mathbf{g}_1[\boldsymbol{\phi}]\right) + \frac{\alpha^2}{2}\left(\frac{\partial\mathbf{g}[\boldsymbol{\phi}]}{\partial\boldsymbol{\phi}}\frac{d\boldsymbol{\phi}}{dt} + \alpha\frac{\partial\mathbf{g}_1[\boldsymbol{\phi}]}{\partial\boldsymbol{\phi}}\frac{d\boldsymbol{\phi}}{dt}\right)\bigg|_{\boldsymbol{\phi}=\boldsymbol{\phi}_0} \\
&= \boldsymbol{\phi} + \alpha\left(\mathbf{g}[\boldsymbol{\phi}] + \alpha\mathbf{g}_1[\boldsymbol{\phi}]\right) + \frac{\alpha^2}{2}\left(\frac{\partial\mathbf{g}[\boldsymbol{\phi}]}{\partial\boldsymbol{\phi}}\mathbf{g}[\boldsymbol{\phi}] + \alpha\frac{\partial\mathbf{g}_1[\boldsymbol{\phi}]}{\partial\boldsymbol{\phi}}\mathbf{g}[\boldsymbol{\phi}]\right)\bigg|_{\boldsymbol{\phi}=\boldsymbol{\phi}_0} \\
&\approx \boldsymbol{\phi} + \alpha\mathbf{g}[\boldsymbol{\phi}] + \alpha^2\left(\mathbf{g}_1[\boldsymbol{\phi}] + \frac{1}{2}\frac{\partial\mathbf{g}[\boldsymbol{\phi}]}{\partial\boldsymbol{\phi}}\mathbf{g}[\boldsymbol{\phi}]\right)\bigg|_{\boldsymbol{\phi}=\boldsymbol{\phi}_0},
\end{aligned} \tag{2.13}$$

- Equation 9.19 Should be total derivative not partial:

$$\begin{aligned}
\frac{d\boldsymbol{\phi}}{dt} &\approx \mathbf{g}[\boldsymbol{\phi}] + \alpha\mathbf{g}_1[\boldsymbol{\phi}] \\
&= -\frac{\partial L}{\partial\boldsymbol{\phi}} - \frac{\alpha}{2}\left(\frac{\partial^2 L}{\partial\boldsymbol{\phi}^2}\right)\frac{\partial L}{\partial\boldsymbol{\phi}}.
\end{aligned} \tag{2.14}$$

- Page 156 Notes: Wrong marginal reference — Appendix B.3.7 Spectral Norm
- Figure 10.3 legend – In dilated or atrous convolution (from the French "à trous" – with holes), we intersperse zeros in the weight vector...
- Section 10.5.1 A final max-pooling layer yields a 6×6 representation with 256 channels which is resized into a vector of length 9,216 and passed through three fully connected layers containing 4096, 4096, and 1000 hidden units, respectively.
- Section 10.5.1 The complete network contains ∼60 million parameters, most of which are in the fully connected layers ~~and the end of the network~~.
- 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. You may assume that the input is padded with zeros at positions x_{-2}, x_{-1} and x_0.
- Equation 11.3: Terms should be reordered to be consistent with definition of vector derivative in appendix:

$$\frac{\partial \mathbf{y}}{\partial \mathbf{f}_1} = \frac{\partial \mathbf{f}_2}{\partial \mathbf{f}_1}\frac{\partial \mathbf{f}_3}{\partial \mathbf{f}_2}\frac{\partial \mathbf{f}_4}{\partial \mathbf{f}_3}. \tag{2.15}$$

- Equation 11.6: Terms should be reordered to be consistent with definition of vector derivative in appendix:

$$\frac{\partial \mathbf{y}}{\partial \mathbf{f}_1} = \mathbf{I} + \frac{\partial \mathbf{f}_2}{\partial \mathbf{f}_1} + \left(\frac{\partial \mathbf{f}_3}{\partial \mathbf{f}_1} + \frac{\partial \mathbf{f}_2}{\partial \mathbf{f}_1}\frac{\partial \mathbf{f}_3}{\partial \mathbf{f}_2}\right) + \left(\frac{\partial \mathbf{f}_4}{\partial \mathbf{f}_1} + \frac{\partial \mathbf{f}_2}{\partial \mathbf{f}_1}\frac{\partial \mathbf{f}_4}{\partial \mathbf{f}_2} + \frac{\partial \mathbf{f}_3}{\partial \mathbf{f}_1}\frac{\partial \mathbf{f}_4}{\partial \mathbf{f}_3} + \frac{\partial \mathbf{f}_2}{\partial \mathbf{f}_1}\frac{\partial \mathbf{f}_3}{\partial \mathbf{f}_2}\frac{\partial \mathbf{f}_4}{\partial \mathbf{f}_3}\right), \tag{2.16}$$

- Equation 11.10:

$$\begin{aligned} f_1 &= \mathbb{E}[z_i] & f_5 &= \sqrt{f_4 + \epsilon} \\ f_{2i} &= z_i - f_1 & f_6 &= 1/f_5 \\ f_{3i} &= f_{2i}^2 & f_{7i} &= f_{2i} \times f_6 \\ f_4 &= \mathbb{E}[f_{3i}] & z_i' &= f_{7i} \times \gamma + \delta, \end{aligned} \tag{2.17}$$

- Figure 12.10 legend: A small fraction of the input tokens are randomly replaced with...
- Section 12.2.1 Not technically wrong, but for consistency with previous and following sections: Equation 12.2 shows that the same weights $\boldsymbol{\Omega}_v \in \mathbb{R}^{D\times D}$ and biases $\boldsymbol{\beta}_v \in \mathbb{R}^D$ are applied to each input $\mathbf{x}_\bullet \in \mathbb{R}^D$...
- Section 12.2.1 Not technically, wrong, but for consistency with previous and following sections: It follows that the number of attention weights has a quadratic dependence on the sequence length N, but is independent of the length D of each input $\cancel{\mathbf{x}_\bullet}$.

- Section 12.3.1 Each element of the attention matrix corresponds to a particular offset between key position a and query position b.
- Section 12.3.2 title: Scaled dot-product self-attention
- Section 12.3.2 This is known as *scaled dot-product self-attention*.
- Problem 12.1 How many weights and biases would there be in a fully connected shallow network relating all DN inputs to all DN outputs?
- Notes Page 236 Much subsequent work has modified just the attention matrix so that in the scaled dot product self-attention equation:
- Problem 12.10: Extra bracket removed:

$$a[\mathbf{x}_m, \mathbf{x}_n] = \text{softmax}_m \left[\mathbf{k}_\bullet^T \mathbf{q}_n\right] = \frac{\exp\left[\mathbf{k}_m^T \mathbf{q}_n\right]}{\sum_{m'=1}^{N} \exp\left[\mathbf{k}_{m'}^T \mathbf{q}_n \cancel{)}\right]}. \tag{2.18}$$

- Figure 13.10 Right hand column should be labelled as "output" not "hidden layer two". item Equation 13.22

$$\mathbf{H}'_k = \boldsymbol{\beta}_k \mathbf{1}^T + \boldsymbol{\Omega}_k \mathbf{H}_k. \tag{2.19}$$

- Section 15.1 A single new sample $\mathbf{x}_j^*$ is generated by (i) choosing a *latent variable* $\mathbf{z}_j$ from a simple base distribution (e.g., a standard normal) and then (ii) passing this data through a network $\mathbf{x}_j{}^* = \mathbf{g}[\mathbf{z}_j, \boldsymbol{\theta}]$ with parameters $\boldsymbol{\theta}$
- Section 15.2.1 If we divide the two sums in the first line of equation 15.5 by the numbers...
- Section 15.2.1 When $I = J$, the optimal discriminator for an example $\tilde{\mathbf{x}}$ of unknown origin is:
- Equation 16.12:

$$\text{f}[h, \boldsymbol{\phi}] = \left(\sum_{k=1}^{b-1} \phi_k\right) + (hK - b)\phi_b, \tag{2.20}$$

- Equation 16.25:

$$\hat{\boldsymbol{\phi}} = \underset{\boldsymbol{\phi}}{\text{argmin}} \left[\text{KL}\left[\frac{1}{I}\sum_{i=1}^{I} \delta\left[\mathbf{x} - \text{f}[\mathbf{z}_i, \boldsymbol{\phi}]\right] \middle|\middle| q(\mathbf{x})\right]\right]. \tag{2.21}$$

- Section 16.2 The first term is the inverse of the determinant of the $D \times D$ Jacobian matrix $\partial\mathbf{f}[\mathbf{z}, \boldsymbol{\phi}]/\partial\mathbf{z}$, which contains elements $\partial\text{f}_i[\mathbf{z}, \boldsymbol{\phi}]/\partial z_j$ at position (i, j).
- Section 16.3.2 ...where the parameters $\phi_1, \phi_2, \ldots, \phi_K$ are positive and sum to 1, and $b = \lfloor Kh \rfloor$ is the index of the bin that contains h.

- Section 17.5 we can't evaluate the evidence term $Pr(\mathbf{x}|\boldsymbol{\phi})$ in the denominator (see section 17.3).
- Equation 17.28 Brackets in third line now larger.
- Section 18.2 This is a *Markov chain* because the probability of $\mathbf{z}_t$ is determined entirely by the value of the immediately preceding variable $\mathbf{z}_{t-1}$.
- Section 18.6.1 The obvious architectural choice for this image-to-image mapping is the U-Net (figure 11.10).
- Section 18.7 Hence, the decision transformer replaces the reward r_t with the *returns-to-go* $R_{t:T} = \sum_{t'=t}^{T} r_{t'}$ (i.e., the sum of the empirically observed future rewards).
- Section 19.1.2 Rephrased as ambiguous: A *Markov reward process* extends the Markov process to include a distribution $Pr(r_{t+1}|s_t)$ over the possible rewards r_{t+1} received at the next time step, given that we are in state s_t.
- Section 20.5.1 In general, the smaller the model, the larger the proportion of weights that can... Note that this statement is only true for pure pruning methods, and not for lottery tickets where the pruned network is retrained from scratch, and so it has been removed.
- Section C.2.1 Rule 2:

$$
\begin{aligned}
\mathbb{E}\Big[k \cdot \mathrm{f}[x]\Big] &= \int k \cdot \mathrm{f}[x] Pr(x) dx \\
&= k \cdot \int \mathrm{f}[x] Pr(x) dx \\
&= k \cdot \mathbb{E}\Big[\mathrm{f}[x]\Big].
\end{aligned}
$$

- Section C.5.4 Reformulated to reflect the fact that the Fréchet/ and 2-Wasserstein distances are the same:

  The Fréchet/2-Wasserstein distance is given by:

$$
D^2_{Fr/W_2}\Big[\mathrm{Norm}[\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1]\Big|\Big|\mathrm{Norm}[\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2]\Big] = |\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2|^2 + \mathrm{tr}\Big[\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2 - 2\left(\boldsymbol{\Sigma}_1\boldsymbol{\Sigma}_2\right)^{1/2}\Big]. \tag{2.22}
$$

First printing (Dec. 2023)

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 2.1 of this document.
- Section 6.3.1 where now the gradients are evaluated at $\boldsymbol{\phi}_t - \alpha\beta \cdot \mathbf{m}_t$.
- Section 7.5.1 The expectation (mean) $\mathbb{E}[f_i{}']$ of the intermediate values $f_i{}'$ is:
- Equation 15.7 The optimal discriminator for an example $\tilde{\mathbf{x}}$ depends on the underlying probabilities:

$$Pr(\text{real}|\tilde{\mathbf{x}}) = \text{sig}\big[\text{f}[\tilde{\mathbf{x}}, \boldsymbol{\phi}]\big] = \frac{Pr(\tilde{\mathbf{x}}|\text{real})}{Pr(\tilde{\mathbf{x}}|\text{generated}) + Pr(\tilde{\mathbf{x}}|\text{real})} = \frac{Pr(\mathbf{x})}{Pr(\mathbf{x}^*) + Pr(\mathbf{x})}. \tag{2.23}$$

  where on the right hand side, we evaluate $\tilde{\mathbf{x}}$ against the generated distribution $Pr(\mathbf{x}^*)$ and the real distribution $Pr(\mathbf{x})$.

- Equation 15.9. First integrand should be with respect to $\mathbf{x}^*$. Correct version is:

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

- Equation 15.12.

$$D_w\Big[Pr(x)||q(x)\Big] = \max_{\mathbf{f}}\left[\sum_i Pr(x=i)f_i - \sum_j q(x=j)f_j\right], \tag{2.24}$$

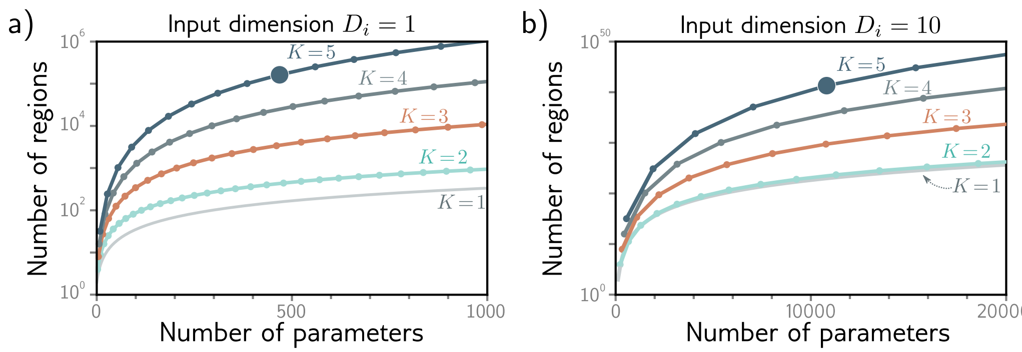


Figure 2.1 Corrected version of figure 4.7: The maximum number of linear regions for neural networks increases rapidly with the network depth. a) Network with $D_i = 1$ input. Each curve represents a fixed number of hidden layers K, as we vary the number of hidden units D per layer. For a fixed parameter 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.

- 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:

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

- 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:

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

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

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

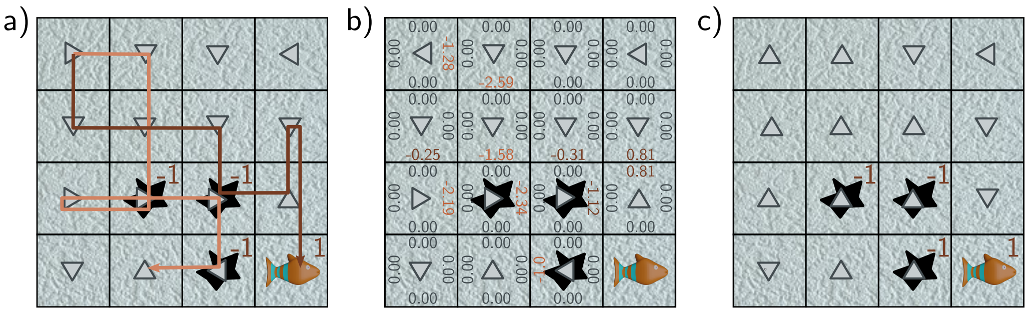


Figure 2.2 Corrected version of figure 19.11

- 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.

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

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

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

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

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 that are chosen so that they alternate the sign of their slope (fourth linear region is outside range of graph).
- Figure 4.2: Colors changed to avoid ambiguity
- 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 5.12. More properly written as:

$$\hat{y} = \underset{y}{\operatorname{argmax}}\left[Pr(y|\mathrm{f}[\mathbf{x},\hat{\boldsymbol{\phi}},\sigma^2])\right]. \tag{2.25}$$

although the value of σ^2 does not actually matter or change the position of the maximum.

- 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 5.5 The likelihood that input $\mathbf{x}$ has label $y=k$ (figure 5.10) is hence:

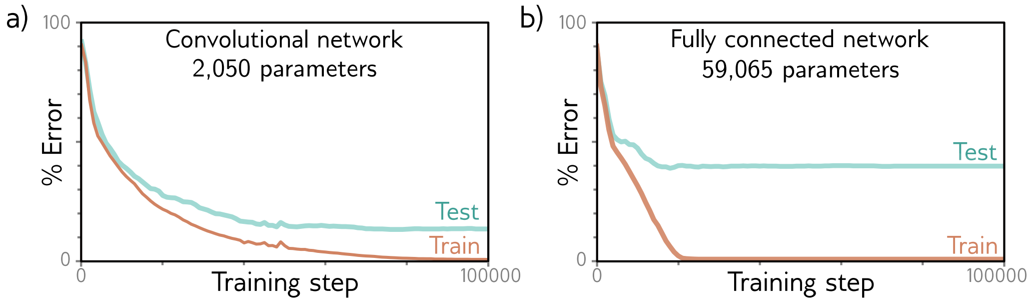
- Section 5.6 Removed i index from this paragraph for consistency. Independence implies that we treat the probability $Pr(\mathbf{y}|\mathbf{f}[\mathbf{x},\boldsymbol{\phi}])$ as a product of univariate terms for each element $y_d \in \mathbf{y}$:

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

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…

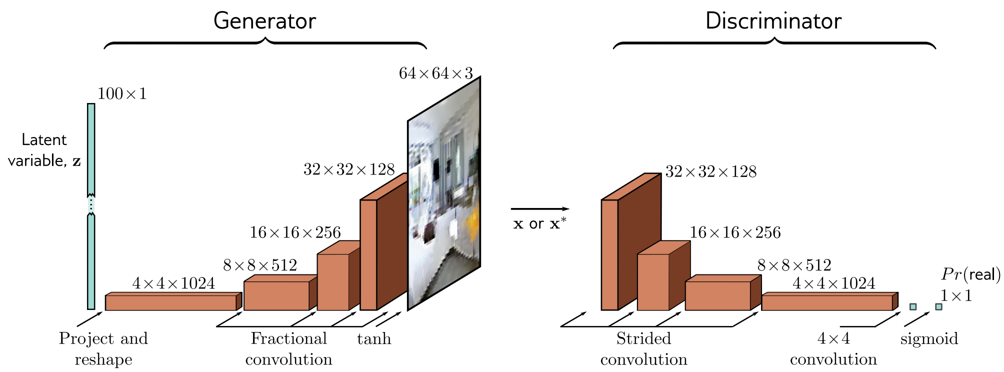
- Notes page 94. However, this is strange since SGD is a special case of Adam (when $\beta = \gamma = 0$)
- Section 7.3. The final derivatives from the term $f_0 = \beta_0 + \omega_0 \cdot x_i$ are:
- Section 7.4. Similarly, the derivative for the weights matrix $\boldsymbol{\Omega}_k$, is given by
- Section 7.5.1 and the second moment $\mathbb{E}[h_j^2]$ will be half the variance σ_f^2
- Figure 7.8. Not wrong, but changed to "nn.init.kaiming_normal_(layer_in.weight)" for compatability with text and to avoid deprecated warning.
- Section 8.3.3 (i.e., with four hidden units and four linear regions in the range of the data) + minor changes in text to accommodate extra words
- Figure 8.9 number of hidden units / linear regions in range of data
- Section 8.4.1 When the number of parameters is very close to the number of training data examples (figure 8.11b)
- Figure 9.5 legend: Effect of learning rate (LR) and batch size for 4000 training and 4000 test examples from MNIST-1D (see figure 8.1) for a neural network with two hidden layers. a) Performance is better for large learning rates than for intermediate or small ones. In each case, the number of iterations is 6000/LR, so each solution has the opportunity to move the same distance.
- 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 10.2.1 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.
- Section 10.2.7 The convolutional network has 2,050 parameters, and the fully connected network has 59,065 parameters.
- Figure 10.8 Number of parameters also wrong in figure 10.8 (correct version in this document). Recalculated curve is slightly different.
- 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.



Corrected version of figure 10.8

- Problem 10.1 Show that the operation in equation 10.3 is equivariant with respect to translation.
- 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: Not a mistake, but this is clearer: where $\boldsymbol{\beta}_v \in \mathbb{R}^D \times 1$ 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 Title changed to Transformer layers
- Section 12.4 a larger transformer mechanism

- 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 legend: 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
- Section 13.5.1 Given I training graphs $\{\mathbf{X}_i, \mathbf{A}_i\}$ and their labels y_i, the parameters $\boldsymbol{\Phi} = \{\boldsymbol{\beta}_k, \boldsymbol{\Omega}_k\}_{k=0}^{K}$ can be learned using SGD...
- Figure 15.3 legend: At the end is a tanh function that maps the...
- Figure 15.3 arctan → tanh. Corrected version nearby in this document.
- Section 15.1.3: At the final layer, the 64×64×3 signal is passed through a tanh function to generate an image $\mathbf{x}^*$
- Equation 15.6. Minor problems with brackets in this equation. Should be:



Corrected version of figure 15.3

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

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

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

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

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

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

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

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

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

- Equation 18.34 missing indices on noise term:

$$
\begin{aligned}
L[\boldsymbol{\phi}_{1\ldots T}] \;=\; & \sum_{i=1}^{I} -\log\Big[\text{Norm}_{\mathbf{x}_i}\left[\mathbf{f}_1[\mathbf{z}_{i1},\boldsymbol{\phi}_1],\sigma_1^2\mathbf{I}\right]\Big] \qquad (2.26)\\
& + \sum_{t=2}^{T}\frac{1}{2\sigma_t^2}\left\|\left(\frac{1}{\sqrt{1-\beta_t}}\mathbf{z}_{it} - \frac{\beta_t}{\sqrt{1-\alpha_t}\sqrt{1-\beta_t}}\boldsymbol{\epsilon}_{it}\right) - \mathbf{f}_t[\mathbf{z}_{it},\boldsymbol{\phi}_t]\right\|^2.
\end{aligned}
$$

- Section 20.2.2 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.

- Section 20.2.4 For example, Du et al. (2019a) show that residual networks converge to zero training loss when the width of the network D (i.e., the number of hidden units) is $\Omega[I^4K^2]$ where I is the amount of training data, and K is the depth of the network.

- Section 21.7 the Conference on AI, Ethics, and Society

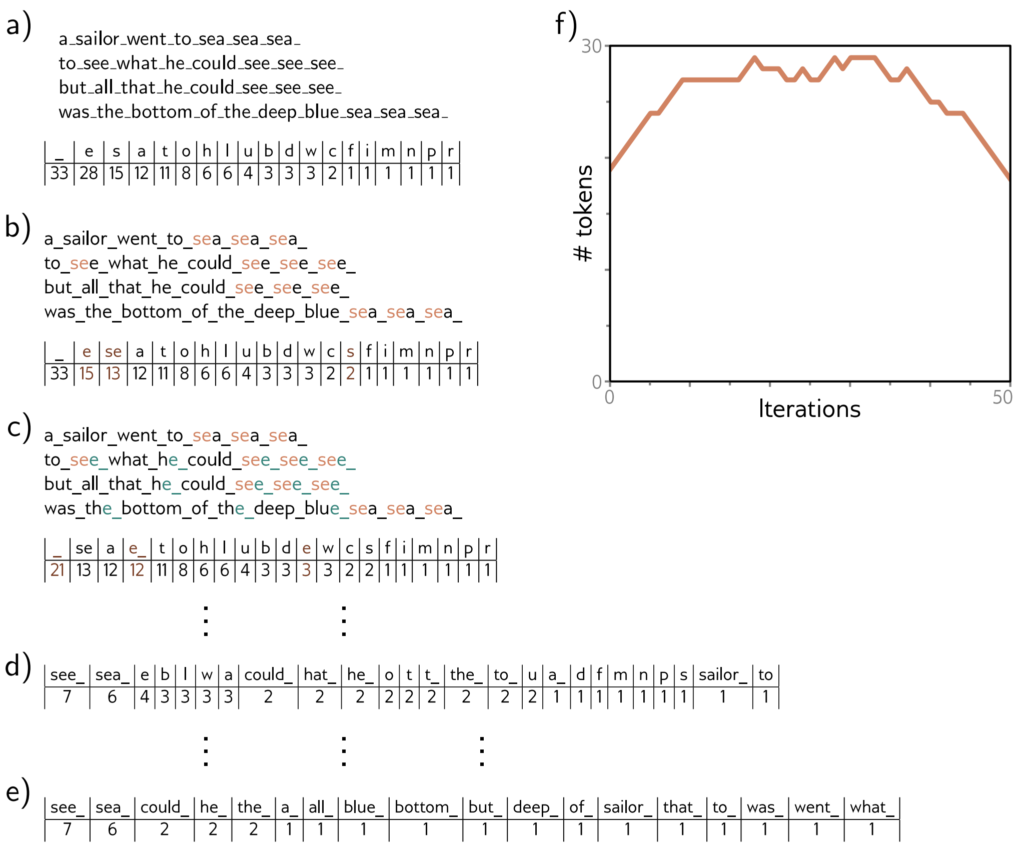
- Appendix A. The notation $\{0, 1, 2, \ldots\}$ denotes the set of non-negative integers.

- Appendix A ...*big-O* notation, which represents an upper bound...

- Appendix A. $\mathrm{f}[n] < c\cdot\mathrm{g}[n]$ for all $n > n_0$

- Equation B. 18

$$
\begin{aligned}
y_1 &= \phi_{10} + \phi_{11}z_1 + \phi_{12}z_2 + \phi_{13}z_3\\
y_2 &= \phi_{20} + \phi_{21}z_1 + \phi_{22}z_2 + \phi_{23}z_3\\
y_3 &= \phi_{30} + \phi_{31}z_1 + \phi_{32}z_2 + \phi_{33}z_3. \qquad (2.27)
\end{aligned}
$$



Corrected version of figure 12.8

- Appendix C.5.4 Accent in wrong place: The Fréchet and Wasserstein distances...
- Equation C.32.

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