Solutions to Numerical Problems in Deep Learning (All Units)

UNIT–I Numerical Problems

Problem 1

Problem. A perceptron receives inputs $(x_1, x_2) = (2, -1)$ with weights $(w_1, w_2) = (0.5, -0.25)$ and bias $b = 0.2$. Compute the perceptron output using the step activation.

Solution. The net input:

$$net = w_1 x_1 + w_2 x_2 + b = 0.5 \cdot 2 + (-0.25)(-1) + 0.2 = 1.0 + 0.25 + 0.2 = 1.45.$$

Using step activation (assuming output 1 if $net \geq 0$, else 0):

$$y = \text{step}(1.45) = 1.$$

So, the perceptron output is $\boxed{1}$.

Problem 2

Problem. For a dataset containing points belonging to two classes, a perceptron misclassifies $(x_1, x_2) = (1, -2)$ with desired output $y = 1$. If learning rate $\eta = 0.1$ and current weights are $(0.4, -0.3)$, compute the updated weights.

Solution. Perceptron learning rule:

$$\mathbf{w}_{\text{new}} = \mathbf{w}_{\text{old}} + \eta(y - \hat{y})\mathbf{x}.$$

Since the example is misclassified and desired output is $y = 1$, we can assume $\hat{y} = -1$ (for $\{-1, +1\}$ activation).

Then:

$$y - \hat{y} = 1 - (-1) = 2, \quad \eta(y - \hat{y}) = 0.1 \times 2 = 0.2.$$

Update:

$$\Delta\mathbf{w} = 0.2 \cdot (1, -2) = (0.2, -0.4).$$

So:

$$w_1^{\text{new}} = 0.4 + 0.2 = 0.6, \quad w_2^{\text{new}} = -0.3 - 0.4 = -0.7.$$

Hence updated weights are $\boxed{(0.6, -0.7)}$.

Problem 3

Problem. A sigmoid neuron computes

$$z = w_1 x_1 + w_2 x_2 + b.$$

Given $x_1 = 1$, $x_2 = 3$, $w_1 = 0.2$, $w_2 = -0.1$, $b = 0.5$, compute z and $\sigma(z)$, where

$$\sigma(z) = \frac{1}{1 + e^{-z}}.$$

Solution. First:

$$z = 0.2 \cdot 1 + (-0.1) \cdot 3 + 0.5 = 0.2 - 0.3 + 0.5 = 0.4.$$

Now:

$$\sigma(0.4) = \frac{1}{1 + e^{-0.4}} \approx 0.5987.$$

So $z = 0.4$, $\sigma(z) \approx \boxed{0.599}$.

Problem 4

Problem. For the loss function

$$L = \frac{1}{2}(t - y)^2,$$

where $y = wx$, $t = 4$, $x = 2$, and $w = 1.5$, compute $\dfrac{\partial L}{\partial w}$.

Solution. We have:

$$y = wx = 1.5 \cdot 2 = 3, \quad t - y = 4 - 3 = 1.$$

Loss:

$$L = \frac{1}{2}(1)^2 = 0.5.$$

Now:

$$\frac{\partial L}{\partial w} = \frac{\partial}{\partial w}\left[\frac{1}{2}(t - wx)^2\right] = (t - wx)(-x) = -(t - y)x = (y - t)x.$$

Substitute:

$$\frac{\partial L}{\partial w} = (3 - 4) \cdot 2 = -2.$$

So $\boxed{\dfrac{\partial L}{\partial w} = -2}$.

Problem 5

Problem. Given learning rate $\eta = 0.01$, update the weight using gradient descent from Q4.

Solution. Gradient descent update:

$$w_{\text{new}} = w_{\text{old}} - \eta \frac{\partial L}{\partial w}.$$

From Q4: $w_{\text{old}} = 1.5$, $\frac{\partial L}{\partial w} = -2$.

$$w_{\text{new}} = 1.5 - 0.01 \cdot (-2) = 1.5 + 0.02 = 1.52.$$

So $\boxed{w_{\text{new}} = 1.52}$.

UNIT–II Numerical Problems

Problem 1

Problem. A network has output $y = \sigma(wx)$ where σ is sigmoid. Given $x = 0.5$, $w = 0.8$, and target $t = 1$, compute the gradient $\dfrac{\partial L}{\partial w}$ for squared error

$$L = \frac{1}{2}(t - y)^2.$$

Solution. First compute:

$$z = wx = 0.8 \cdot 0.5 = 0.4, \quad y = \sigma(z) \approx 0.5987.$$

Now:

$$\frac{\partial L}{\partial y} = y - t = 0.5987 - 1 = -0.4013.$$

Also:

$$\frac{\partial y}{\partial z} = y(1 - y) \approx 0.5987(1 - 0.5987) \approx 0.2403,$$

and

$$\frac{\partial z}{\partial w} = x = 0.5.$$

Chain rule:

$$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial y}\frac{\partial y}{\partial z}\frac{\partial z}{\partial w} \approx (-0.4013)(0.2403)(0.5) \approx -0.0482.$$

So $\boxed{\dfrac{\partial L}{\partial w} \approx -0.0482}$.

Problem 2

Problem. In Momentum-based GD, compute the new velocity and weight:

$$v_t = \beta v_{t-1} + (1 - \beta)\nabla L, \quad w_t = w_{t-1} - \eta v_t$$

Given $\beta = 0.9$, $v_{t-1} = 0.4$, $\nabla L = 0.2$, $\eta = 0.01$, $w_{t-1} = 2$.

Solution.

$$v_t = 0.9 \cdot 0.4 + 0.1 \cdot 0.2 = 0.36 + 0.02 = 0.38.$$

$$w_t = 2 - 0.01 \cdot 0.38 = 2 - 0.0038 = 1.9962.$$

So $\boxed{v_t = 0.38,\ w_t = 1.9962}$.

Problem 3

Problem. In RMSProp, compute the updated running average:

$$E[g^2]_t = 0.9E[g^2]_{t-1} + 0.1g_t^2$$

Given $E[g^2]_{t-1} = 0.5$, $g_t = 0.3$.

Solution.

$$E[g^2]_t = 0.9 \cdot 0.5 + 0.1 \cdot (0.3)^2 = 0.45 + 0.1 \cdot 0.09 = 0.45 + 0.009 = 0.459.$$

So $\boxed{E[g^2]_t = 0.459}$.

Problem 4

Problem. Compute PCA: Given covariance matrix

$$C = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix},$$

compute eigenvalues.

Solution. Solve $\det(C - \lambda I) = 0$:

$$\det \begin{bmatrix} 2-\lambda & 1 \\ 1 & 2-\lambda \end{bmatrix} = (2-\lambda)^2 - 1 = 0.$$

So:

$$(2-\lambda)^2 = 1 \Rightarrow 2 - \lambda = \pm 1.$$

Hence:

$$\lambda_1 = 2 - 1 = 1, \quad \lambda_2 = 2 + 1 = 3.$$

So eigenvalues are $\boxed{\lambda_1 = 1,\ \lambda_2 = 3}$.

Problem 5

Problem. Perform SVD on

$$A = \begin{bmatrix} 3 & 0 \\ 4 & 5 \end{bmatrix}.$$

Find $A^T A$.

Solution.

$$A^T = \begin{bmatrix} 3 & 4 \\ 0 & 5 \end{bmatrix}, \quad A^T A = \begin{bmatrix} 3 & 4 \\ 0 & 5 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 4 & 5 \end{bmatrix} = \begin{bmatrix} 3\cdot 3 + 4\cdot 4 & 3\cdot 0 + 4\cdot 5 \\ 0\cdot 3 + 5\cdot 4 & 0\cdot 0 + 5\cdot 5 \end{bmatrix} = \begin{bmatrix} 9+16 & 20 \\ 20 & 25 \end{bmatrix} = \begin{bmatrix} 25 & 20 \\ 20 & 25 \end{bmatrix}.$$

So $\boxed{A^T A = \begin{bmatrix} 25 & 20 \\ 20 & 25 \end{bmatrix}}$.

UNIT–III Numerical Problems

Problem 1

Problem. A denoising autoencoder input is $x = 0.8$ and noise $\epsilon = -0.2$. Compute the corrupted input $\tilde{x}$.

Solution.

$$\tilde{x} = x + \epsilon = 0.8 + (-0.2) = 0.6.$$

So $\boxed{\tilde{x} = 0.6}$.

Problem 2

Problem. In L2 regularization, compute the regularized loss:

$$L_{\text{reg}} = L + \lambda \|w\|^2,$$

where $L = 0.4$, $\lambda = 0.01$, $\|w\|^2 = 25$.

Solution.

$$L_{\text{reg}} = 0.4 + 0.01 \cdot 25 = 0.4 + 0.25 = 0.65.$$

So $\boxed{L_{\text{reg}} = 0.65}$.

Problem 3

Problem. A dropout layer drops 40% of neurons. If input activation sum is 12, compute the expected activation after dropout during training (without rescaling).

Solution. Keep probability $p = 1 - 0.4 = 0.6$. Expected activation:

$$\mathbb{E}[\text{activation after dropout}] = p \times 12 = 0.6 \times 12 = 7.2.$$

So $\boxed{7.2}$.

Problem 4

Problem. Batch Normalization transforms

$$\hat{x} = \frac{x - \mu}{\sigma}, \quad y = \gamma \hat{x} + \beta.$$

Given $x = 10$, $\mu = 8$, $\sigma = 2$, $\gamma = 1.5$, $\beta = 0.5$, compute y.

Solution.

$$\hat{x} = \frac{10 - 8}{2} = \frac{2}{2} = 1.$$

Then:

$$y = 1.5 \cdot 1 + 0.5 = 1.5 + 0.5 = 2.0.$$

So $\boxed{y = 2.0}$.

Problem 5

Problem. A sparse autoencoder has reconstruction error 0.12 and sparsity penalty 0.03. Compute total cost.

Solution.

$$\text{Total cost} = 0.12 + 0.03 = 0.15.$$

So $\boxed{0.15}$.

UNIT–IV Numerical Problems

Problem 1

Problem. Perform 1D convolution: Input: $[1, 2, 3, 4]$, Filter: $[1, -1]$, Stride = 1. Compute output (valid convolution).

Solution. Windows and outputs:

$$[1, 2] \to 1 \cdot 1 + 2 \cdot (-1) = 1 - 2 = -1,$$

$$[2, 3] \to 2 \cdot 1 + 3 \cdot (-1) = 2 - 3 = -1,$$

$$[3, 4] \to 3 \cdot 1 + 4 \cdot (-1) = 3 - 4 = -1.$$

So output is $\boxed{[-1,\ -1,\ -1]}$.

Problem 2

Problem. Given a 2×2 pooling window with max pooling on

$$\begin{bmatrix} 1 & 3 \\ 2 & 6 \end{bmatrix}$$

compute pooled value.

Solution. The maximum value:

$$\max\{1, 3, 2, 6\} = 6.$$

So pooled value is $\boxed{6}$.

Problem 3

Problem. A CNN layer uses 32 filters of size $3 \times 3 \times 16$. Compute total trainable parameters (including bias).

Solution. Each filter has:

$$3 \times 3 \times 16 = 144 \text{ weights.}$$

Including 1 bias per filter:

$$\text{params per filter} = 144 + 1 = 145.$$

For 32 filters:

$$\text{Total parameters} = 145 \times 32 = 4640.$$

So $\boxed{4640}$ parameters.

Problem 4

Problem. In ResNet, a residual block has input $x = 5$ and $F(x) = -2$. Compute block output $y = F(x) + x$.

Solution.

$$y = F(x) + x = -2 + 5 = 3.$$

So $\boxed{y = 3}$.

Problem 5

Problem. A feature map has dimension $28 \times 28 \times 64$. After applying 2×2 stride-2 max pooling, compute output size.

Solution. Pooling with window 2×2 and stride 2 halves height and width:

$$28 \to 14, \quad 28 \to 14, \quad \text{depth unchanged (64).}$$

So output size: $\boxed{14 \times 14 \times 64}$.

UNIT–V Numerical Problems

Problem 1

Problem. An RNN computes

$$h_t = \tanh(Wh_{t-1} + Ux_t)$$

Given $h_{t-1} = 0.5$, $x_t = 1$, $W = 0.2$, $U = 0.4$, compute h_t.

Solution. Pre-activation:

$$a_t = Wh_{t-1} + Ux_t = 0.2 \cdot 0.5 + 0.4 \cdot 1 = 0.1 + 0.4 = 0.5.$$

Thus:

$$h_t = \tanh(0.5) \approx 0.4621.$$

So $\boxed{h_t \approx 0.462}$.

Problem 2

Problem. In BPTT, total gradient is

$$\frac{\partial L}{\partial w} = \sum_{t=1}^{3} \frac{\partial L_t}{\partial w}.$$

If $0.3, 0.5, 0.2$ are gradients for $t = 1, 2, 3$, compute the total.

Solution.

$$\frac{\partial L}{\partial w} = 0.3 + 0.5 + 0.2 = 1.0.$$

So $\boxed{1.0}$.

Problem 3

Problem. In LSTM, forget gate is

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f).$$

Given $x_t = 1$, $h_{t-1} = 2$, $W_f = 0.5$, $U_f = -0.1$, $b_f = 0$, compute f_t.

Solution.

$$z = W_f x_t + U_f h_{t-1} + b_f = 0.5 \cdot 1 + (-0.1) \cdot 2 + 0 = 0.5 - 0.2 = 0.3.$$

$$f_t = \sigma(0.3) = \frac{1}{1 + e^{-0.3}} \approx 0.5744.$$

So $\boxed{f_t \approx 0.574}$.

Problem 4

Problem. In attention mechanism, compute attention score

$$e = qk$$

Given $q = 0.6$, $k = 0.4$.

Solution.

$$e = qk = 0.6 \cdot 0.4 = 0.24.$$

So $\boxed{e = 0.24}$.

Problem 5

Problem. Compute softmax values for

$$z = [2, 1, 0]$$

to three decimal places.

Solution. Softmax:

$$\alpha_i = \frac{e^{z_i}}{\sum_j e^{z_j}}.$$

Here:

$$e^2,\ e^1,\ e^0 = 7.389,\ 2.718,\ 1.$$

Sum:

$$S = 7.389 + 2.718 + 1 \approx 11.107.$$

Thus:

$$\alpha_1 = \frac{7.389}{11.107} \approx 0.665, \quad \alpha_2 = \frac{2.718}{11.107} \approx 0.245, \quad \alpha_3 = \frac{1}{11.107} \approx 0.090.$$

So:

$$\boxed{(0.665,\ 0.245,\ 0.090)}.$$

Attention Scores based Problems

Problem 1

Problem. Given a query vector

$$q = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad k_1 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \quad k_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

compute the unnormalized attention scores

$$e_1 = q^T k_1, \quad e_2 = q^T k_2.$$

Solution.

$$e_1 = q^T k_1 = 1 \cdot 2 + 2 \cdot 0 = 2,$$
$$e_2 = q^T k_2 = 1 \cdot 1 + 2 \cdot 1 = 1 + 2 = 3.$$

So $\boxed{e_1 = 2,\ e_2 = 3}$.

Problem 2

Problem. Using the scores from Question 1, compute the attention weights

$$\alpha_i = \frac{\exp(e_i)}{\exp(e_1) + \exp(e_2)}, \quad i = 1, 2.$$

Express α_1 and α_2 numerically (approximate up to 3 decimal places).

Solution. We have $e_1 = 2$, $e_2 = 3$:

$$\alpha_1 = \frac{e^2}{e^2 + e^3} = \frac{1}{1 + e}, \quad \alpha_2 = \frac{e^3}{e^2 + e^3} = \frac{e}{1 + e}.$$

Numerically:

$$\alpha_1 \approx 0.269, \quad \alpha_2 \approx 0.731.$$

So $\boxed{\alpha_1 \approx 0.269,\ \alpha_2 \approx 0.731}$.

Problem 3

Problem. Let the value vectors be

$$v_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad v_2 = \begin{bmatrix} 0 \\ 2 \end{bmatrix}.$$

Using the attention weights α_1, α_2 from Question 2, compute the context vector

$$c = \alpha_1 v_1 + \alpha_2 v_2.$$

Solution.

$$c = \alpha_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \alpha_2 \begin{bmatrix} 0 \\ 2 \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 2\alpha_2 \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ 2\alpha_2 \end{bmatrix}.$$

Using numerical values:

$$c \approx \begin{bmatrix} 0.269 \\ 2 \times 0.731 \end{bmatrix} = \begin{bmatrix} 0.269 \\ 1.462 \end{bmatrix}.$$

So $\boxed{c \approx [0.269,\ 1.462]^T}$.

Problem 4

Problem. Consider scaled dot-product attention. Given

$$q = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \quad k = \begin{bmatrix} 1 \\ 3 \end{bmatrix}, \quad d_k = 2,$$

compute the scaled attention score

$$e = \frac{q^T k}{\sqrt{d_k}}.$$

Solution.

$$q^T k = 2 \cdot 1 + 1 \cdot 3 = 2 + 3 = 5, \quad \sqrt{d_k} = \sqrt{2}.$$

Thus:

$$e = \frac{5}{\sqrt{2}} \approx 3.536.$$

So $\boxed{e \approx 3.536}$.

Problem 5

Problem. You are given three keys and one query:

$$q = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad k_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad k_2 = \begin{bmatrix} 2 \\ -1 \end{bmatrix}, \quad k_3 = \begin{bmatrix} 0 \\ 2 \end{bmatrix}.$$

(a) Compute the unnormalized scores $e_i = q^T k_i$ for $i = 1, 2, 3$.
(b) Compute the softmax attention weights

$$\alpha_i = \frac{\exp(e_i)}{\sum_{j=1}^{3} \exp(e_j)}, \quad i = 1, 2, 3.$$

Solution. (a)

$$e_1 = q^T k_1 = 1 \cdot 1 + 0 \cdot 1 = 1,$$
$$e_2 = q^T k_2 = 1 \cdot 2 + 0 \cdot (-1) = 2,$$
$$e_3 = q^T k_3 = 1 \cdot 0 + 0 \cdot 2 = 0.$$

(b) Softmax over $[1, 2, 0]$:

$$\alpha_i = \frac{e^{e_i}}{e^1 + e^2 + e^0}.$$

Using approximate values:

$$e^1 = 2.718, \quad e^2 = 7.389, \quad e^0 = 1, \quad S \approx 11.107.$$

Then:

$$\alpha_1 \approx \frac{2.718}{11.107} \approx 0.245,$$
$$\alpha_2 \approx \frac{7.389}{11.107} \approx 0.665,$$
$$\alpha_3 \approx \frac{1}{11.107} \approx 0.090.$$

So $\boxed{\alpha \approx (0.245,\ 0.665,\ 0.090)}$.

Problem 6

Problem. Given attention logits (unnormalized scores)

$$e = [2, 1, 0],$$

compute the softmax attention weights

$$\alpha_i = \frac{\exp(e_i)}{\exp(2) + \exp(1) + \exp(0)}, \quad i = 1, 2, 3.$$

Provide the values up to 3 decimal places.

Solution. Same as UNIT–V Problem 5:

$$\alpha \approx (0.665,\ 0.245,\ 0.090).$$

So $\boxed{\alpha_1 \approx 0.665,\ \alpha_2 \approx 0.245,\ \alpha_3 \approx 0.090}$.

Problem 7

Problem. Masked attention: suppose we have scores

$$e = [3, -1, 0.5]$$

and we apply a mask that disallows the second position by assigning it $-\infty$. Conceptually, this is implemented as:

$$e' = [3, -\infty, 0.5].$$

(a) Write the softmax expression for $\alpha_1, \alpha_2, \alpha_3$ using e'.
(b) Explain why $\alpha_2 = 0$ and compute approximate values for α_1 and α_3 (up to 3 decimal places).

Solution. (a) Softmax with masked scores:

$$\alpha_i = \frac{\exp(e'_i)}{\sum_{j=1}^{3} \exp(e'_j)}, \quad i = 1, 2, 3.$$

So:

$$\alpha_1 = \frac{e^3}{e^3 + e^{-\infty} + e^{0.5}}, \quad \alpha_2 = \frac{e^{-\infty}}{e^3 + e^{-\infty} + e^{0.5}}, \quad \alpha_3 = \frac{e^{0.5}}{e^3 + e^{-\infty} + e^{0.5}}.$$

(b) Since $e^{-\infty} = 0$, we get:

$$\alpha_2 = 0,$$

and effectively:

$$\alpha_1 = \frac{e^3}{e^3 + e^{0.5}}, \quad \alpha_3 = \frac{e^{0.5}}{e^3 + e^{0.5}}.$$

Numerically:

$$e^3 \approx 20.086, \quad e^{0.5} \approx 1.649, \quad S \approx 21.735.$$

Thus:

$$\alpha_1 \approx \frac{20.086}{21.735} \approx 0.924, \quad \alpha_3 \approx \frac{1.649}{21.735} \approx 0.076.$$

So:

$$\boxed{\alpha_1 \approx 0.924,\ \alpha_2 = 0,\ \alpha_3 \approx 0.076}.$$

Problem 8

Problem. Multi-head style small example: Let the query and key matrices be

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad K = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Compute the score matrix

$$S = QK^T.$$

Write out S explicitly.

Solution. First:

$$K^T = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Then:

$$S = QK^T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

So $\boxed{S = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}}$.

Problem 9

Problem. For a single query q and two key-value pairs (k_1, v_1) and (k_2, v_2), attention is defined as:

$$\text{Attention}(q, K, V) = \alpha_1 v_1 + \alpha_2 v_2,$$

where

$$\alpha_i = \frac{\exp(q^T k_i)}{\exp(q^T k_1) + \exp(q^T k_2)}.$$

Given:

$$q = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad k_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad k_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad v_1 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \quad v_2 = \begin{bmatrix} 0 \\ 3 \end{bmatrix},$$

compute the context vector $\text{Attention}(q, K, V)$ numerically.

Solution. First:

$$q^T k_1 = 1 \cdot 1 + 1 \cdot 0 = 1, \quad q^T k_2 = 1 \cdot 0 + 1 \cdot 1 = 1.$$

Hence:

$$\alpha_1 = \alpha_2 = \frac{e^1}{e^1 + e^1} = \frac{1}{2}.$$

Context:

$$\text{Attention}(q, K, V) = \frac{1}{2}\begin{bmatrix} 2 \\ 0 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} 0 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1.5 \end{bmatrix} = \begin{bmatrix} 1 \\ 1.5 \end{bmatrix}.$$

So $\boxed{\text{Attention}(q, K, V) = [1,\ 1.5]^T}$.

Problem 10

Problem. Scaled dot-product attention in matrix form: Let

$$Q = \begin{bmatrix} 1 & 2 \end{bmatrix}, \quad K = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad V = \begin{bmatrix} 2 & 1 \\ 3 & 0 \end{bmatrix}, \quad d_k = 2.$$

(a) Compute the score vector

$$s = \frac{QK^T}{\sqrt{d_k}}.$$

(b) Apply softmax to s to get attention weights α.
(c) Compute the final output

$$\text{output} = \alpha V.$$

Solution. (a) Since K is the identity, $K^T = I$ and:

$$QK^T = Q = [1\ \ 2].$$

Thus:

$$s = \frac{[1\ \ 2]}{\sqrt{2}} = \left[\frac{1}{\sqrt{2}}, \frac{2}{\sqrt{2}}\right] \approx [0.7071,\ 1.4142].$$

(b) Softmax over s:

$$\alpha_i = \frac{e^{s_i}}{e^{s_1} + e^{s_2}}.$$

Numerically:

$$e^{0.7071} \approx 2.028, \quad e^{1.4142} \approx 4.113, \quad S \approx 6.141.$$

So:

$$\alpha_1 \approx \frac{2.028}{6.141} \approx 0.330, \quad \alpha_2 \approx \frac{4.113}{6.141} \approx 0.670.$$

(c) Now $\alpha = [\alpha_1 \;\; \alpha_2]$:

$$\text{output} = \alpha V = [\alpha_1 \;\; \alpha_2] \begin{bmatrix} 2 & 1 \\ 3 & 0 \end{bmatrix}.$$

Compute components:

$$\text{output}_1 = 2\alpha_1 + 3\alpha_2 \approx 2(0.330) + 3(0.670) \approx 0.660 + 2.010 = 2.670,$$

$$\text{output}_2 = 1\alpha_1 + 0\alpha_2 \approx 0.330.$$

So:

$$\text{output} \approx \boxed{[2.670,\ 0.330]}.$$