Discussion 10

Note: *Your TA will probably not cover all the problems on this worksheet. The discussion worksheets are not designed to be finished within an hour. They are deliberately made slightly longer so they can serve as resources you can use to practice, reinforce, and build upon concepts discussed in lectures, discussions, and homework.*

This Week's Cool AI Demo/Video:

1 Query-Key-Value in Transformer Attention

The attention mechanism was a key building block introduced by the paper *Attention Is All You Need* in 2017 that has jump-started unprecedented advances in deep learning architectures. Attention helps machine learning models determine the relative importance of each part of an input sequence to other parts of the input sequence. In this problem, we will see how queries, keys, and values are calculated in the attention process and how they allow models to *attend* to their inputs.

Assume that the encoder in our transformer is processing the embeddings of three tokens:

$$\mathbf{x}_1 = \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \quad \mathbf{x}_2 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \quad \mathbf{x}_3 = \begin{bmatrix} 1 \\ -2 \end{bmatrix}. \qquad N=3,\ D=2$$

Our attention layer has the following weight matrices:

$$\mathbf{W_Q} = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & -3 \end{bmatrix}, \quad \mathbf{W_K} = \begin{bmatrix} 0 & 0 & -1 \\ -2 & 0 & 1 \end{bmatrix}, \quad \mathbf{W}_V = \begin{bmatrix} 8 & 0 \\ 0 & 9 \end{bmatrix}. \qquad D_k=3,\ D_v=2$$

The query, key, and value of each embedding vector are defined respectively as

$$\mathbf{q}_i = \mathbf{W}_Q^\top \mathbf{x}_i, \qquad \mathbf{k}_i = \mathbf{W}_K^\top \mathbf{x}_i, \qquad \mathbf{v}_i = \mathbf{W}_V^\top \mathbf{x}_i.$$

(a) Compute the query, key, and value vectors for $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$.

(b) Recall that the attention mechanism in transformers allows the model to decide how much each token should "focus" on every other token in the sequence. To do this, the model computes an attention score for every *ordered* pair of tokens by first taking the dot product between the query vector of one token and the key vector of another. For example, the inner product between the i-th token's query, $\mathbf{q}_i$, and the j-th token's key, $\mathbf{k}_j$, is: $z_{i,j} = \mathbf{q}_i^T \mathbf{k}_j$.

The inner product $z_{i,j}$ measures how stoken $\mathbf{x}_i$ should consider or "pay attention" to token $\mathbf{x}_j$. Calculate the attention that token $\mathbf{x}_3$ places on each of the three tokens, $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$.

(c) We would like to transform these attention scores into probabilities, so we will apply the softmax function. Taking the softmax over $z_{3,1}$, $z_{3,2}$, and $z_{3,3}$, we obtain the following probabilities:

$$a_{3,1} \approx 0.00247, \qquad a_{3,2} \approx 0.99753, \qquad a_{3,3} \approx 6.90 \times 10^{-13}.$$

The resulting softmax scores act as weights for forming a weighted sum of the value vectors. Write an expression for this weighted sum for $\mathbf{x}_3$ and plug in the values you computed previously.

(d) Transformers benefit from efficient matrix operations. To parallelize our computations, we stack all our input embeddings, $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$, into the rows of a data matrix $\mathbf{X}$:

$$\mathbf{X} = \begin{bmatrix} -\mathbf{x}_1^T - \\ -\mathbf{x}_2^T - \\ -\mathbf{x}_3^T - \end{bmatrix} = \begin{bmatrix} 2 & 3 \\ -1 & 0 \\ 1 & -2 \end{bmatrix}.$$

We want to obtain query, key, and value *matrices*, where

$$\mathbf{Q} = \begin{bmatrix} -\mathbf{q}_1^T - \\ -\mathbf{q}_2^T - \\ -\mathbf{q}_3^T - \end{bmatrix}, \quad \mathbf{K} = \begin{bmatrix} -\mathbf{k}_1^T - \\ -\mathbf{k}_2^T - \\ -\mathbf{k}_3^T - \end{bmatrix}, \quad \mathbf{V} = \begin{bmatrix} -\mathbf{v}_1^T - \\ -\mathbf{v}_2^T - \\ -\mathbf{v}_3^T - \end{bmatrix}.$$

Write an expression for these three matrices in terms of the data matrix, $\mathbf{X}$, and the weight matrices defined in the problem statement. What are the dimensions of these matrices?

(e) Using the $\mathbf{Q}$ and $\mathbf{K}$ matrices from the previous step, show that the (i, j)-th entry of the matrix product $\mathbf{QK}^T$ is exactly $z_{i,j}$ from part (b).

(f) The i-th row in the matrix product $\mathbf{QK}^T$ represents how much $\mathbf{x}_i$ should attend to every other token $\mathbf{x}_j$. Recall that we apply the softmax over the attention scores $z_{i,1}$, $z_{i,2}$, $z_{i,3}$ to turn them into probabilities. In matrix world, this means we apply the softmax to each row of $\mathbf{QK}^T$, such that all the entries are non-negative and the entries in each row sum to 1.

Let $\mathbf{A}$ be the result of applying the softmax function to $\mathbf{QK}^T$ row-wise. Show that the i-th row of $\mathbf{AV}$ is the weighted sum of the value vectors for $\mathbf{x}_i$, using the softmax scores as weights.

Assume that the encoder in our transformer is processing the embeddings of three tokens:

$$\mathbf{x}_1 = \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \qquad \mathbf{x}_2 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \qquad \mathbf{x}_3 = \begin{bmatrix} 1 \\ -2 \end{bmatrix}.$$

2x1 2x1 2x1

Our attention layer has the following weight matrices:

$$\mathbf{W_Q} = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & -3 \end{bmatrix}, \qquad \mathbf{W_K} = \begin{bmatrix} 0 & 0 & -1 \\ -2 & 0 & 1 \end{bmatrix}, \qquad \mathbf{W}_V = \begin{bmatrix} 8 & 0 \\ 0 & 9 \end{bmatrix}.$$

2x3 2x3 2x2

The query, key, and value of each embedding vector are defined respectively as

$$\mathbf{q}_i = \mathbf{W}_Q^\top \mathbf{x}_i, \qquad \mathbf{k}_i = \mathbf{W}_K^\top \mathbf{x}_i, \qquad \mathbf{v}_i = \mathbf{W}_V^\top \mathbf{x}_i.$$

3x1 3x2 2x1 3x1 3x2 2x1 2x1 2x2 2x1

(a) Compute the query, key, and value vectors for $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$.

query: $q_i = W_Q^T X_i$

$$q_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 2 & -3 \end{pmatrix}\begin{pmatrix} 2 \\ 3 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ -5 \end{pmatrix} \qquad q_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 2 & -3 \end{pmatrix}\begin{pmatrix} -1 \\ 0 \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \\ 2 \end{pmatrix}$$

$$q_3 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 2 & -3 \end{pmatrix}\begin{pmatrix} 1 \\ -2 \end{pmatrix} = \begin{pmatrix} 1 \\ -2 \\ 8 \end{pmatrix}$$

key: $k_i = W_k^T X_i$

$$k_1 = \begin{pmatrix} 0 & -2 \\ 0 & 0 \\ -1 & 1 \end{pmatrix}\begin{pmatrix} 2 \\ 3 \end{pmatrix} = \begin{pmatrix} -6 \\ 0 \\ 1 \end{pmatrix} \qquad k_2 = \begin{pmatrix} 0 & -2 \\ 0 & 0 \\ -1 & 1 \end{pmatrix}\begin{pmatrix} -1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$k_3 = \begin{pmatrix} 0 & -2 \\ 0 & 0 \\ -1 & 1 \end{pmatrix}\begin{pmatrix} 1 \\ -2 \end{pmatrix} = \begin{pmatrix} 4 \\ 0 \\ -3 \end{pmatrix}$$

Value: $V_i = W_V^T X_i$

$$V_1 = \begin{bmatrix} 8 & 0 \\ 0 & 9 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 16 \\ 27 \end{bmatrix} \qquad V_2 = \begin{bmatrix} 8 & 0 \\ 0 & 9 \end{bmatrix} \begin{bmatrix} -1 \\ 0 \end{bmatrix} = \begin{bmatrix} -8 \\ 0 \end{bmatrix}$$

$$V_3 = \begin{bmatrix} 8 & 0 \\ 0 & 9 \end{bmatrix} \begin{bmatrix} 1 \\ -2 \end{bmatrix} = \begin{bmatrix} 8 \\ -18 \end{bmatrix}$$

(b) Recall that the attention mechanism in transformers allows the model to decide how much each token should "focus" on every other token in the sequence. To do this, the model computes an attention score for every *ordered* pair of tokens by first taking the dot product between the query vector of one token and the key vector of another. For example, the inner product between the i-th token's query, $\mathbf{q}_i$, and the j-th token's key, $\mathbf{k}_j$, is: $z_{i,j} = \mathbf{q}_i^T \mathbf{k}_j$.

The inner product $z_{i,j}$ measures how much token $\mathbf{x}_i$ should consider or "pay attention" to token $\mathbf{x}_j$. Calculate the attention that token $\mathbf{x}_3$ places on each of the three tokens, $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$.

Similarity: $z_{i,j} = q_i^T k_j$

$$z_{3,1} = q_3^T k_1 = \begin{bmatrix} 1 & -2 & 8 \end{bmatrix} \begin{bmatrix} -6 \\ 0 \\ 1 \end{bmatrix} = 2$$

$$z_{3,2} = q_3^T k_2 = \begin{bmatrix} 1 & -2 & 8 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = 8$$

$$z_{3,3} = q_3^T k_3 = \begin{bmatrix} 1 & -2 & 8 \end{bmatrix} \begin{bmatrix} 4 \\ 0 \\ -3 \end{bmatrix} = -20$$

(c) We would like to transform these attention scores into probabilities, so we will apply the softmax function. Taking the softmax over $z_{3,1}$, $z_{3,2}$, and $z_{3,3}$, we obtain the following probabilities:

$$a_{3,1} \approx 0.00247, \qquad a_{3,2} \approx 0.99753, \qquad a_{3,3} \approx 6.90 \times 10^{-13}.$$

The resulting softmax scores act as weights for forming a weighted sum of the value vectors. Write an expression for this weighted sum for $\mathbf{x}_3$ and plug in the values you computed previously.

$$\sum_{j=1}^{3} a_{3,j} v_j = 0.00247 \begin{bmatrix} 16 \\ 27 \end{bmatrix} + 0.99753 \begin{bmatrix} -8 \\ 0 \end{bmatrix} + 6.90 \times 10^{-13} \begin{bmatrix} 8 \\ -18 \end{bmatrix} = \begin{bmatrix} -7.94 \\ 0.067 \end{bmatrix}$$

(d) Transformers benefit from efficient matrix operations. To parallelize our computations, we stack all our input embeddings, $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$, into the rows of a data matrix $\mathbf{X}$:

$$\mathbf{X} = \begin{bmatrix} -\mathbf{x}_1^T - \\ -\mathbf{x}_2^T - \\ -\mathbf{x}_3^T - \end{bmatrix} = \begin{bmatrix} 2 & 3 \\ -1 & 0 \\ 1 & -2 \end{bmatrix}.$$

3x2

We want to obtain query, key, and value *matrices*, where

$$\mathbf{Q} = \begin{bmatrix} -\mathbf{q}_1^T - \\ -\mathbf{q}_2^T - \\ -\mathbf{q}_3^T - \end{bmatrix}, \quad \mathbf{K} = \begin{bmatrix} -\mathbf{k}_1^T - \\ -\mathbf{k}_2^T - \\ -\mathbf{k}_3^T - \end{bmatrix}, \quad \mathbf{V} = \begin{bmatrix} -\mathbf{v}_1^T - \\ -\mathbf{v}_2^T - \\ -\mathbf{v}_3^T - \end{bmatrix}.$$

Write an expression for these three matrices in terms of the data matrix, $\mathbf{X}$, and the weight matrices defined in the problem statement. What are the dimensions of these matrices?

query: $$Q = \begin{bmatrix} -q_1^T - \\ -q_2^T - \\ -q_3^T - \end{bmatrix} = \begin{bmatrix} -(W_Q^T x_1)^T - \\ -(W_Q^T x_2)^T - \\ -(W_Q^T x_3)^2 - \end{bmatrix} = \begin{bmatrix} -x_1^T W_Q - \\ -x_2^T W_Q - \\ -x_3^T W_Q - \end{bmatrix}$$

$$= \begin{bmatrix} -x_1^T - \\ -x_2^T - \\ -x_3^T - \end{bmatrix} W_Q = \underset{3\times2}{X} \underset{2\times3}{W_Q} \in \mathbb{R}^{3\times3} \text{ or } \mathbb{R}^{N \times D_k}$$

key: $K = \begin{bmatrix} -k_1^T - \\ -k_2^T - \\ -k_3^T - \end{bmatrix} = \begin{bmatrix} -(W_K^T x_1)^T - \\ -(W_K^T x_2)^T - \\ -(W_K^T x_3)^2 - \end{bmatrix} = \begin{bmatrix} - x_1^T W_K - \\ - x_2^T W_K - \\ - x_3^T W_K - \end{bmatrix}$

$= \begin{bmatrix} -x_1^T - \\ -x_2^T - \\ -x_3^T - \end{bmatrix} W_K = \underset{3\times 2}{X} \underset{2\times 3}{W_K} \in \mathbb{R}^{3\times 3}$ or $\mathbb{R}^{N\times D_K}$

value: $V = \begin{bmatrix} -v_1^T - \\ -v_2^T - \\ -v_3^T - \end{bmatrix} = \begin{bmatrix} -(W_V^T x_1)^T - \\ -(W_V^T x_2)^T - \\ -(W_V^T x_3)^2 - \end{bmatrix} = \begin{bmatrix} - x_1^T W_V - \\ - x_2^T W_V - \\ - x_3^T W_V - \end{bmatrix}$

$= \begin{bmatrix} -x_1^T - \\ -x_2^T - \\ -x_3^T - \end{bmatrix} W_V = \underset{3\times 2}{X} \underset{2\times 2}{W_V} \in \mathbb{R}^{3\times 2}$ or $\mathbb{R}^{N\times D_V}$

(e) Using the $\mathbf{Q}$ and $\mathbf{K}$ matrices from the previous step, show that the (i, j)-th entry of the matrix product $\mathbf{QK}^T$ is exactly $z_{i,j}$ from part (b).

$$QK^T = \begin{bmatrix} -q_1^T - \\ -q_2^T - \\ -q_3^T - \end{bmatrix} \begin{bmatrix} -k_1^T - \\ -k_2^T - \\ -k_3^T - \end{bmatrix}^T = \begin{bmatrix} -q_1^T - \\ -q_2^T - \\ -q_3^T - \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ k_1 & k_2 & k_3 \\ 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} q_1^T k_1 & q_1^T k_2 & q_1^T k_3 \\ q_2^T k_1 & q_2^T k_2 & q_2^T k_3 \\ q_3^T k_1 & q_3^T k_2 & q_3^T k_3 \end{bmatrix} \quad (QK^T)_{ij} = q_i^T k_j = z_{i,j}$$

(f) The i-th row in the matrix product $\mathbf{QK}^T$ represents how much $\mathbf{x}_i$ should attend to every other token $\mathbf{x}_j$. Recall that we apply the softmax over the attention scores $z_{i,1}$, $z_{i,2}$, $z_{i,3}$ to turn them into probabilities. In matrix world, this means we apply the softmax to each row of $\mathbf{QK}^T$, such that all the entries are non-negative and the entries in each row sum to 1.

Let $\mathbf{A}$ be the result of applying the softmax function to $\mathbf{QK}^T$ row-wise. Show that the i-th row of $\mathbf{AV}$ is the weighted sum of the value vectors for $\mathbf{x}_i$, using the softmax scores as weights.

$$AV = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} \begin{bmatrix} v_{1,1} & v_{1,2} \\ v_{2,1} & v_{2,2} \\ v_{3,1} & v_{3,2} \end{bmatrix}$$

$$= \begin{bmatrix} a_{1,1}v_{1,1} + a_{1,2}v_{2,1} + a_{1,3}v_{3,1} & a_{1,1}v_{1,2} + a_{1,2}v_{2,2} + a_{1,3}v_{3,2} \\ a_{2,1}v_{1,1} + a_{2,2}v_{2,1} + a_{2,3}v_{3,1} & a_{2,1}v_{1,2} + a_{2,2}v_{2,2} + a_{2,3}v_{3,2} \\ a_{3,1}v_{1,1} + a_{3,2}v_{2,1} + a_{3,3}v_{3,1} & a_{3,1}v_{1,2} + a_{3,2}v_{2,2} + a_{3,3}v_{3,2} \end{bmatrix}$$

i^{th} row of AV:

$$(AV)_i = \begin{bmatrix} a_{i,1}v_{1,1} + a_{i,2}v_{2,1} + a_{i,3}v_{3,1} & a_{i,1}v_{1,2} + a_{i,2}v_{2,2} + a_{i,3}v_{3,2} \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{j=1}^{3} a_{i,j}v_{j,1} & \sum_{j=1}^{3} a_{i,j}v_{j,2} \end{bmatrix}$$

$$= \sum_{j=1}^{3} a_{i,j} \begin{bmatrix} v_{j,1} & v_{j,2} \end{bmatrix}$$

$$= \sum_{j=1}^{3} a_{i,j} v_j^T$$

2 Justifying Scaled-Dot Product Attention

In the previous problem, we worked through an example of softmax inner-product self-attention. In transformers, we actually apply **scaled** softmax inner-product self-attention. In this problem we'll explore where this scaling factor comes from.

Suppose $\mathbf{q}, \mathbf{k} \in \mathbb{R}^{D_k}$ are two random vectors with $\mathbf{q}, \mathbf{k} \overset{iid}{\sim} \mathcal{N}(\mu\mathbf{1}, \sigma^2\mathbf{I})$, where $\mu\mathbf{1} \in \mathbb{R}^{D_k}$ and $\sigma \in \mathbb{R}^+$. In other words, each component q_i of $\mathbf{q}$ is drawn from a normal distribution with mean μ and standard deviation σ, and the same is true for k_i of $\mathbf{k}$.

(a) Define $\mathbb{E}[\mathbf{q}^\top \mathbf{k}]$ in terms of μ, σ and D_k.

(b) Considering a practical case where $\mu = 0$ and $\sigma = 1$, define $\text{Var}(\mathbf{q}^\top \mathbf{k})$ in terms of D_k.

(c) Continue to assume $\mu = 0$ and $\sigma = 1$. Let s be the scaling factor on the dot product. Suppose we want $\mathbb{E}\left[\frac{\mathbf{q}^\top \mathbf{k}}{s}\right]$ to be 0, and $\text{Var}\left(\frac{\mathbf{q}^\top \mathbf{k}}{s}\right)$ to be $\sigma = 1$. What should s be in terms of D_k?

Suppose $\mathbf{q}, \mathbf{k} \in \mathbb{R}^{D_k}$ are two random vectors with $\mathbf{q}, \mathbf{k} \overset{iid}{\sim} \mathcal{N}(\mu\mathbf{1}, \sigma^2\mathbf{I})$, where $\mu\mathbf{1} \in \mathbb{R}^{D_k}$ and $\sigma \in \mathbb{R}^+$. In other words, each component q_i of $\mathbf{q}$ is drawn from a normal distribution with mean μ and standard deviation σ, and the same is true for k_i of $\mathbf{k}$.

(a) Define $\mathbb{E}[\mathbf{q}^\top \mathbf{k}]$ in terms of μ, σ and D_k.

$$q^T k = \sum_{i=1}^{D_k} q_i k_i$$

Using the linearity of expectation,

$$E(q^T k) = E\left[\sum_{i=1}^{D_k} q_i k_i\right] = \sum_{i=1}^{D_k} E(q_i k_i)$$

B/c q_i & k_i are independent,

$$E(q_i k_i) = E(q_i)\,E(k_i) = \mu \cdot \mu = \mu^2$$

$$E(q^T k) = \sum_{i=1}^{D_k} E(q_i k_i) = \sum_{i=1}^{D_k} \mu^2 = D_k \mu^2$$

(b) Considering a practical case where $\mu = 0$ and $\sigma = 1$, define $\text{Var}(\mathbf{q}^\top \mathbf{k})$ in terms of D_k.

$\text{Var}(q^T k) = \text{Var}\left(\sum_{i=1}^{D_k} q_i k_i\right)$

B/c $q_1k_1, q_2k_2, \ldots, q_{D_k}k_{D_k}$ are independent,
$\text{Var}(q^T k) = \sum_{i=1}^{D_k} \text{Var}(q_i k_i)$

Using the def'n of variance,
$\text{Var}(q_i k_i) = E[(q_i k_i)^2] - E[q_i k_i]^2$

Again, b/c q_i & k_i are independent,
$E[q_i k_i] = E[q_i]\,E[k_i] = \mu \cdot \mu = \mu^2$
$E[(q_i k_i)^2] = E[q_i^2 k_i^2] = E[q_i^2]\,E[k_i^2]$

Using the def'n of variance again,
$E[q_i^2] = \text{Var}(q_i) + E[q_i]^2 = \sigma^2 + \mu^2$
$E[k_i^2] = \text{Var}(k_i) + E[k_i]^2 = \sigma^2 + \mu^2$

Now assuming $\mu = 0$ & $\sigma = 1$,
$E[q_i k_i] = 0$ $\qquad$ $E[(q_i k_i)^2] = E[q_i^2]\,E[k_i^2] = 1$
$E[q_i^2] = E[k_i]^2 = 1$ $\qquad$ $\text{Var}(q_i k_i) = E[(q_i k_i)^2] - E[q_i k_i]^2 = 1$

$\therefore \text{Var}(q^T k) = \sum_{i=1}^{D_k} \text{Var}(q_i k_i) = \sum_{i=1}^{D_k} 1 = D_k$

(c) Continue to assume $\mu = 0$ and $\sigma = 1$. Let s be the scaling factor on the dot product. Suppose we want $\mathbb{E}\left[\frac{\mathbf{q}^\top \mathbf{k}}{s}\right]$ to be 0, and $\mathrm{Var}\left(\frac{\mathbf{q}^\top \mathbf{k}}{s}\right)$ to be $\sigma = 1$. What should s be in terms of D_k?

Using the linearity of expectation,

$$E\left[\frac{q^T k}{s}\right] = \frac{1}{s} E[q^T k] = \frac{1}{s} D_k \mu^2 = 0$$

($\frac{1}{s} E[q^T k]$: (a); $D_k \mu^2 = 0$: assume: $\mu = 0$)

Using the property $\mathrm{Var}(aX + b) = a^2 \mathrm{Var}(X)$,

$$\mathrm{Var}\left(\frac{q^T k}{s}\right) = \frac{1}{s^2} \mathrm{Var}(q^T k) = \frac{1}{s^2} D_k$$

($\frac{1}{s^2} \mathrm{Var}(q^T k)$: (b) + assumptions)

$E\left[\frac{q^T k}{s}\right] = 0$ regardless of the value of s

$\mathrm{Var}\left(\frac{q^T k}{s}\right) = 1$ if $\frac{1}{s^2} D_k = 1 \equiv s^2 = D_k \equiv s = \pm\sqrt{D_k}$

We'll assume the scaling factor is positive, so $s = \sqrt{D_k}$. This should look familiar!