Homework 4 Review Outline

- Decision Trees
- PCA
- Attention

Decision Trees

Problem 1: Decision Trees (12pts)

Consider a binary dataset with 400 examples, where half of them belong to class A and the rest belong to class B. Next, consider two decision stumps (i.e. trees with depth 1) $\mathcal{T}_1$ and $\mathcal{T}_2$, each with two children. For $\mathcal{T}_1$, the left child has 150 examples in class A and 50 examples in class B. For $\mathcal{T}_2$, the left child has 0 examples in class A and 100 examples in class B. (You can infer the number of examples in the right child using the total number of examples.)

1.1 (6 pts) In class, we discussed entropy and Gini impurity as measures of uncertainty at a leaf. Another possible metric is the classification error at the leaf, assuming that the prediction at the leaf is the majority class among all examples that belong to that leaf. For each leaf of $\mathcal{T}_1$ and $\mathcal{T}_2$, compute the entropy (base e), Gini impurity and classification error. You can either exactly express the final numbers in terms of fractions and logarithms, or round them to two decimal places.

1.2 (6 pts) Compare the quality of $\mathcal{T}_1$ and $\mathcal{T}_2$ (that is, the two different splits of the root) based on conditional entropy (base e), weighted Gini impurity, and total classification error. Intuitively, which of $\mathcal{T}_1$ or $\mathcal{T}_2$ appears to be a better split to you (there may not necessarily be one correct answer to this)? Based on your conditional entropy, Gini impurity, and classification error calculations, which of the metrics appear to be more suitable choices to decide which variable to split on?

Decision Trees

Problem 1: Decision Trees (12pts)

Consider a binary dataset with 400 examples, where half of them belong to class A and the rest belong to class B. Next, consider two decision stumps (i.e. trees with depth 1) $\mathcal{T}_1$ and $\mathcal{T}_2$, each with two children. For $\mathcal{T}_1$, the left child has 150 examples in class A and 50 examples in class B. For $\mathcal{T}_2$, the left child has 0 examples in class A and 100 examples in class B. (You can infer the number of examples in the right child using the total number of examples.)

1.1 (6 pts) In class, we discussed entropy and Gini impurity as measures of uncertainty at a leaf. Another possible metric is the classification error at the leaf, assuming that the prediction at the leaf is the majority class among all examples that belong to that leaf. For each leaf of $\mathcal{T}_1$ and $\mathcal{T}_2$, compute the entropy (base e), Gini impurity and classification error. You can either exactly express the final numbers in terms of fractions and logarithms, or round them to two decimal places.

1.2 (6 pts) Compare the quality of $\mathcal{T}_1$ and $\mathcal{T}_2$ (that is, the two different splits of the root) based on conditional entropy (base e), weighted Gini impurity, and total classification error. Intuitively, which of $\mathcal{T}_1$ or $\mathcal{T}_2$ appears to be a better split to you (there may not necessarily be one correct answer to this)? Based on your conditional entropy, Gini impurity, and classification error calculations, which of the metrics appear to be more suitable choices to decide which variable to split on?

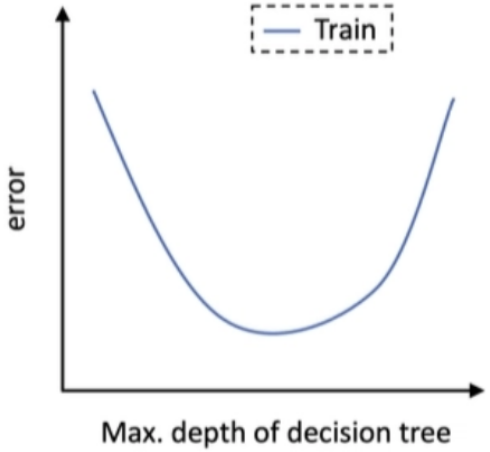
T1 and T2 are as good in terms of classification error. T2 is better in terms of both conditional entropy and weighted Gini impurity. Also, T2 leads to a pure (100% certain) node.

T2 is probably preferable to T1.

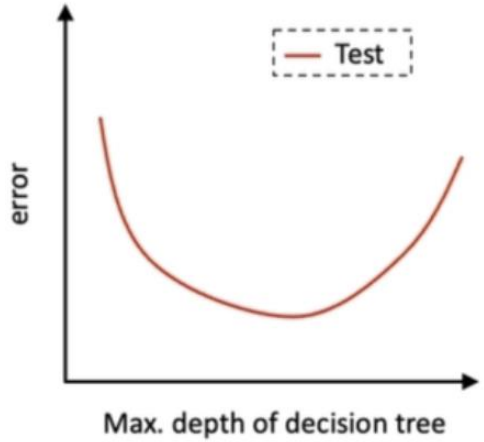
Conditional entropy and Gini impurity appear to be more suitable measures to decide which variable to split on. They are more sensitive to changes in the node probabilities than classification error.

Decision Trees

(6) Figure 3 shows various training/test classification error curves as parameters for training decision trees are varied. Select all options which represent reasonable relationships between the considered parameters and obtained error(s).



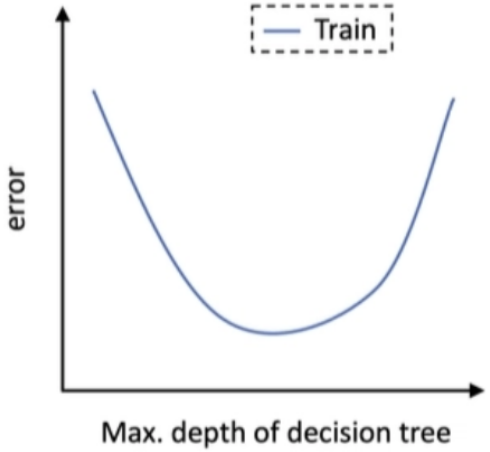
(A) Fig 3a is a reasonable plot of training error as the maximum depth of the decision tree is increased.



(B) Fig 3b is a reasonable plot of test error as the maximum depth of the decision tree is increased.

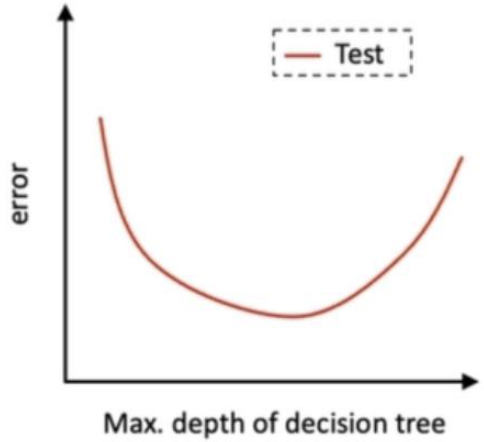
Decision Trees

(6) Figure 3 shows various training/test classification error curves as parameters for training decision trees are varied. Select all options which represent reasonable relationships between the considered parameters and obtained error(s).



(A) Fig 3a is a reasonable plot of training error as the maximum depth of the decision tree is increased.

Incorrect.

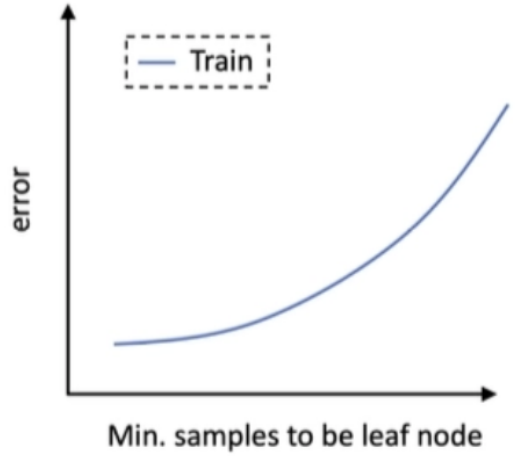


(B) Fig 3b is a reasonable plot of test error as the maximum depth of the decision tree is increased.

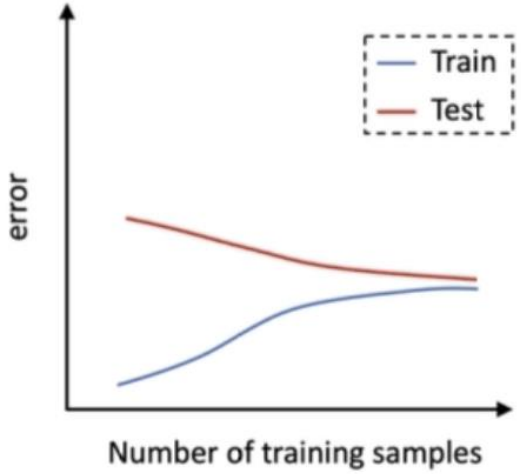
Correct.

Decision Trees

(6) Figure 3 shows various training/test classification error curves as parameters for training decision trees are varied. Select all options which represent reasonable relationships between the considered parameters and obtained error(s).



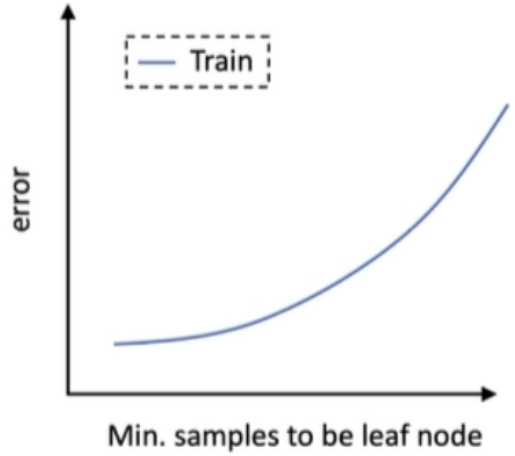
(C) Fig 3c is a reasonable plot of training error as the minimum number of samples to be a leaf node is increased.



(D) Fig 3d is a reasonable plot of training & test errors as the total size of the training set used to train a decision tree with maximum depth 5 is increased.

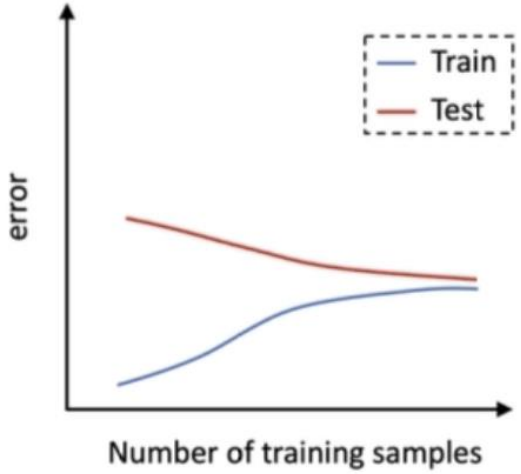
Decision Trees

(6) Figure 3 shows various training/test classification error curves as parameters for training decision trees are varied. Select all options which represent reasonable relationships between the considered parameters and obtained error(s).



(C) Fig 3c is a reasonable plot of training error as the minimum number of samples to be a leaf node is increased.

Correct.



(D) Fig 3d is a reasonable plot of training & test errors as the total size of the training set used to train a decision tree with maximum depth 5 is increased.

Correct.

Decision Trees

Which of the following about decision trees is correct?

(A) Good interpretability is a key advantage of decision trees.
(B) Decision tree algorithms are usually implemented using recursion.
(C) Entropy is the only way to measure the uncertainty of a node when building a decision tree.
(D) Regularization is not applicable to decision trees since they do not minimize a certain loss function.

Decision Trees

Which of the following about decision trees is correct?

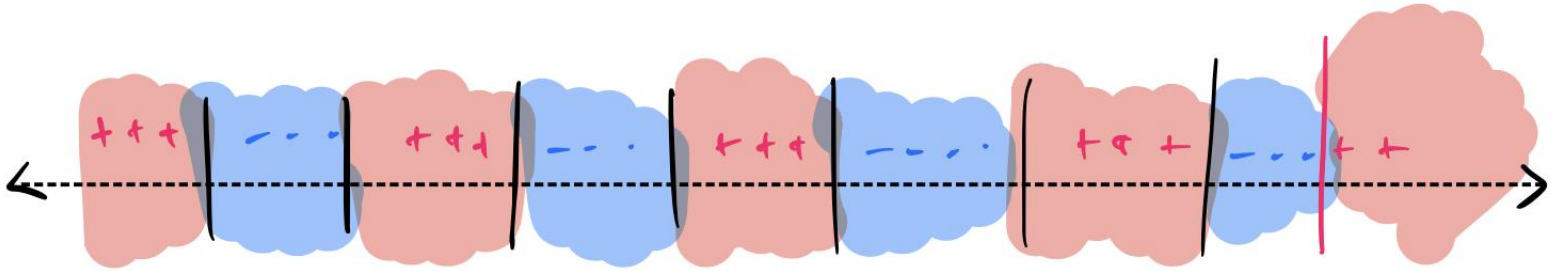
(A) Good interpretability is a key advantage of decision trees.
(B) Decision tree algorithms are usually implemented using recursion.
(C) Entropy is the only way to measure the uncertainty of a node when building a decision tree.
(D) Regularization is not applicable to decision trees since they do not minimize a certain loss function.

Answer: AB

Decision Trees (recap)

Regularization

If the dataset has no contradiction (i.e. same x but different y), the training error of our decision tree algorithm is always zero, and hence the model can overfit.



To prevent overfitting:

- restrict the depth or #nodes (e.g. stop building the tree when the depth reaches some threshold).
- do not split a node if the #examples at the node is smaller than some threshold.
- other approaches as well, all make use of a validation set to tune these hyperparameters

PCA

Which of the following about Principal Component Analysis (PCA) is correct?

(A) PCA can be used to compress a dataset.
(B) PCA is useful for visualizing a dataset.
(C) PCA requires finding all the eigenvectors of the covariance matrix.
(D) To decide how many principal components to use, we can plot some of the eigenvalues and see if they seem to be becoming small.

PCA

Which of the following about Principal Component Analysis (PCA) is correct?

(A) PCA can be used to compress a dataset.
(B) PCA is useful for visualizing a dataset.
(C) PCA requires finding all the eigenvectors of the covariance matrix.
(D) To decide how many principal components to use, we can plot some of the eigenvalues and see if they seem to be becoming small.

Answer: ABD

*For more hands-on PCA-related math problems, see problem 1 of the 2nd mini-discussion (and lecture)!

Attention

For a query $\boldsymbol{q}$, keys $\boldsymbol{k}_1, \ldots, \boldsymbol{k}_n$, and values $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$, the single-headed attention mechanism outputs a weighted combination of the values, i.e., $\boldsymbol{o} = \sum_{i=1}^{n} \alpha_i \boldsymbol{v}_i$, where the weights $\alpha_1, \ldots, \alpha_n$ are given by

$$\alpha_i = \frac{\exp\left(\boldsymbol{q}^T \boldsymbol{k}_i\right)}{\sum_{j=1}^{n} \exp\left(\boldsymbol{q}^T \boldsymbol{k}_j\right)}, \quad \forall i = 1, \ldots, n.$$

This allows the attention mechanism to selectively aggregate information from different values depending on the query. In this problem, you shall explore some fundamental properties of the single-headed attention mechanism.

2.1 (3 pts) **Copying.** The attention mechanism can copy a specific value (say $\boldsymbol{v}_j$) to the output $\boldsymbol{o}$, i.e., $\boldsymbol{o} \approx \boldsymbol{v}_j$. Note that exact copying ($\boldsymbol{o} = \boldsymbol{v}_j$) is not possible with the softmax function since $0 < \alpha_i < 1$ for all i, but approximate copying, i.e. $\boldsymbol{o} \approx \boldsymbol{v}_j$, is achievable. Derive a relationship between the query and keys such that this is possible.

Hint: Try to relate α_j with α_i for $i \neq j$.

2.2 (5 pts) **Averaging.** The attention mechanism can aggregate information equally from two different values $\boldsymbol{v}_a, \boldsymbol{v}_b$, $a \neq b$, so that $\boldsymbol{o} \approx \frac{1}{2}(\boldsymbol{v}_a + \boldsymbol{v}_b)$. Assume that the keys are orthogonal, i.e., $\boldsymbol{k}_i^\intercal \boldsymbol{k}_j = 0$ for all $i \neq j$, and have unit norm ($\|\boldsymbol{k}_i\|_2 = 1$ for all i). Derive an expression for the query $\boldsymbol{q}$ (in terms of the keys) such that this is possible.

Hint: The desired $\boldsymbol{q}$ can be expressed as a linear combination of the keys. You might also want to introduce a large positive constant in the expression for $\boldsymbol{q}$.

Credit: Stanford CS 224n

Attention

$$\alpha_i = \frac{\exp\left(\boldsymbol{q}^T \boldsymbol{k}_i\right)}{\sum_{j=1}^{n} \exp\left(\boldsymbol{q}^T \boldsymbol{k}_j\right)} \qquad \boldsymbol{o} = \sum_{i=1}^{n} \alpha_i \boldsymbol{v}_i$$

Problem 2.1: $\boldsymbol{o} \approx \boldsymbol{v}_j$

This corresponds to $\boldsymbol{q}^\intercal \boldsymbol{k}_j \gg \boldsymbol{q}^\intercal \boldsymbol{k}_i$ for all $i \neq j$, so that $\exp\left(\boldsymbol{q}^\intercal \boldsymbol{k}_j\right) \gg \exp\left(\boldsymbol{q}^\intercal \boldsymbol{k}_i\right)$ and thus $\alpha_j \approx 1$, $\alpha_i \approx 0$ for all $i \neq j$, giving $\boldsymbol{o} \approx \boldsymbol{v}_j$. (3 points)

Problem 2.2: $\boldsymbol{o} \approx \frac{1}{2}(\boldsymbol{v}_a + \boldsymbol{v}_b)$ $\boldsymbol{k}_i^\intercal \boldsymbol{k}_j = 0$ for all $i \neq j$, $\|\boldsymbol{k}_i\|_2 = 1$ for all i

Set $\boldsymbol{q} = \frac{c}{2}(\boldsymbol{k}_a + \boldsymbol{k}_b)$, where c is a large positive constant. (1 points) Since $\boldsymbol{k}_1, \ldots, \boldsymbol{k}_n$ are orthogonal and have unit norm,

$$\boldsymbol{q}^\intercal \boldsymbol{k}_i = \begin{cases} 0 & \text{for } i \notin \{a, b\}, \\ \frac{c}{2} & \text{for } i \in \{a, b\}. \end{cases}$$

(2 points) Thus $\alpha_a = \alpha_b = \frac{\exp(\frac{c}{2})}{2\exp(\frac{c}{2}) + n - 2} \approx \frac{1}{2}$ for sufficiently large c (specifically, $c \gg 2 \ln \frac{n}{2}$ suffices). For $i \notin \{a, b\}$, $\alpha_i \approx 0$. Therefore $\boldsymbol{o} \approx \frac{1}{2}(\boldsymbol{v}_a + \boldsymbol{v}_b)$. (2 points)

Self-Attention

Self-Attention. In natural language, a word often derives its meaning or reference from other words in the same sentence. Consider the sentence *"Mary packed her bags before the trip."* To correctly interpret *her*, a model must link it back to *Mary* — the word *her* must *attend* to *Mary*. Beyond such coreference resolution, the meaning of a word can shift entirely based on context — *bank* means something different when surrounded by *river* and *fishing* versus *loan* and *interest*. In all these cases, understanding a token requires aggregating information from other tokens in the sequence.

Self-attention achieves this by applying the attention mechanism above within a sequence: each token acts as a query and attends over all other tokens, using learned matrices $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$ to determine which tokens to focus on. Formally, let $d > n$ and $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ be an input sequence of token embeddings, where $\boldsymbol{x}_i \in \mathbb{R}^d$. For each token i, the self-attention layer maintains query $\boldsymbol{q}_i = \boldsymbol{Q}\boldsymbol{x}_i$, key $\boldsymbol{k}_i = \boldsymbol{K}\boldsymbol{x}_i$, and value $\boldsymbol{v}_i = \boldsymbol{V}\boldsymbol{x}_i$, with attention weights

$$\alpha_{i,j} = \frac{\exp\left(\boldsymbol{q}_i^\intercal \boldsymbol{k}_j\right)}{\sum_{j'=1}^{n} \exp\left(\boldsymbol{q}_i^\intercal \boldsymbol{k}_{j'}\right)},$$

and output $\boldsymbol{o}_i = \sum_{j=1}^{n} \alpha_{i,j} \boldsymbol{v}_j$. As a simple first instance of token referencing, we say the self-attention layer allows each token to reference only its immediately preceding token if $\boldsymbol{o}_i \approx \boldsymbol{v}_{i-1}$ for all $i \geq 2$, i.e., the output of token i is primarily determined by the value of token $i - 1$.

2.3 (4 pts) Assume $\boldsymbol{x}_i = \boldsymbol{e}_i$ for all i, where $\boldsymbol{e}_i$ is the i-th standard basis vector. Show that there exist matrices $\boldsymbol{Q}$ and $\boldsymbol{K}$ such that $\boldsymbol{o}_i \approx \boldsymbol{v}_{i-1}$ for all $i \geq 2$, and prove your construction is correct. We recommend spending some time trying to figure out the construction on your own first. Then, if you want help you can look at the hint in the footnote here and prove that the construction there is correct.[1]

Credit: Stanford CS 224n

Self-Attention

$$\alpha_{i,j} = \frac{\exp\left(\boldsymbol{q}_i^\intercal \boldsymbol{k}_j\right)}{\sum_{j'=1}^{n} \exp\left(\boldsymbol{q}_i^\intercal \boldsymbol{k}_{j'}\right)},$$

and output $\boldsymbol{o}_i = \sum_{j=1}^{n} \alpha_{i,j} \boldsymbol{v}_j$. As a simple first instance of token referencing, we say the self-attention layer allows each token to reference only its immediately preceding token if $\boldsymbol{o}_i \approx \boldsymbol{v}_{i-1}$ for all $i \geq 2$, i.e., the output of token i is primarily determined by the value of token $i-1$.

2.3 (4 pts) Assume $\boldsymbol{x}_i = \boldsymbol{e}_i$ for all i, where $\boldsymbol{e}_i$ is the i-th standard basis vector. Show that there exist matrices $\boldsymbol{Q}$ and $\boldsymbol{K}$ such that $\boldsymbol{o}_i \approx \boldsymbol{v}_{i-1}$ for all $i \geq 2$, and prove your construction is correct. We recommend spending some time trying to figure out the construction on your own first. Then, if you want help you can look at the hint in the footnote here and prove that the construction there is correct.[1]

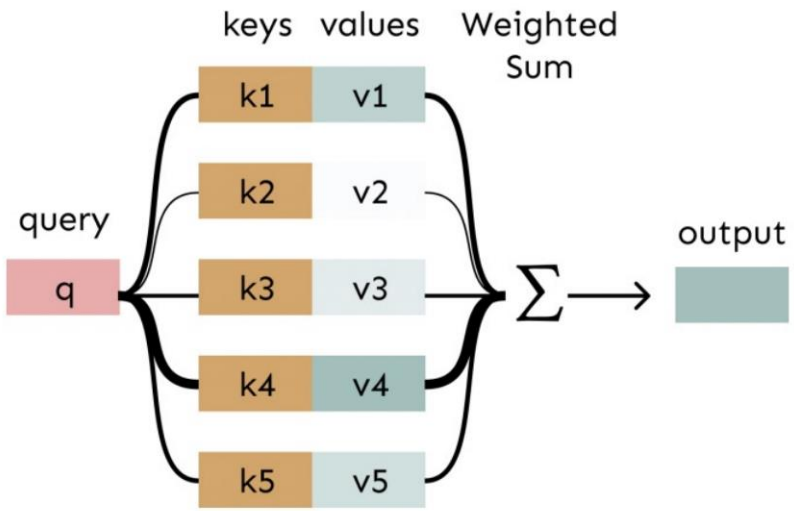
Let c be a large positive constant. Set $\boldsymbol{K} = \boldsymbol{I}$ and $\boldsymbol{Q}$ such that the i-th row of $\boldsymbol{Q}$ is $c\boldsymbol{e}_{i+1}^\intercal$ for $i = 1, \ldots, d-1$ and the d-th row of $\boldsymbol{Q}$ is the zero vector. For $i \geq 2$, we have $\boldsymbol{q}_i = \boldsymbol{Q}\boldsymbol{x}_i = \boldsymbol{Q}\boldsymbol{e}_i = c\,\boldsymbol{e}_{i-1}$ and $\boldsymbol{k}_j = \boldsymbol{K}\boldsymbol{e}_j = \boldsymbol{e}_j$.

$$\boldsymbol{q}_i^\intercal \boldsymbol{k}_j = c\boldsymbol{e}_{i-1}^\intercal \boldsymbol{e}_j = c \cdot \mathbb{I}[j = i-1].$$

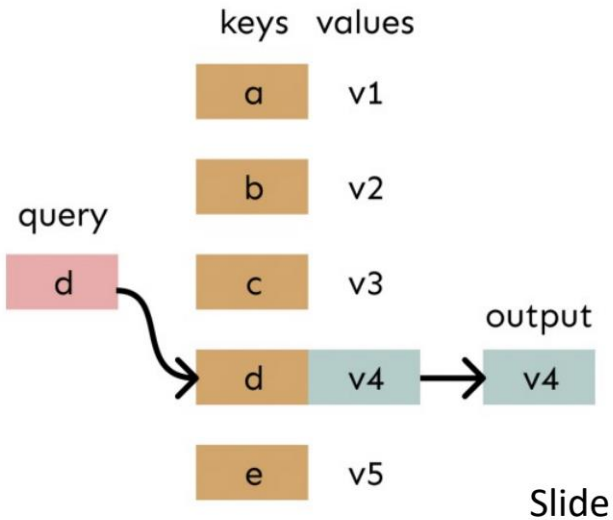
(2 points) Hence $\exp\left(\boldsymbol{q}_i^\intercal \boldsymbol{k}_{i-1}\right) = \exp(c) \gg 1 = \exp\left(\boldsymbol{q}_i^\intercal \boldsymbol{k}_j\right)$ for all $j \neq i-1$, giving $\alpha_{i,i-1} \approx 1$ and $\alpha_{i,j} \approx 0$ for $j \neq i-1$. Therefore $\boldsymbol{o}_i \approx \boldsymbol{v}_{i-1}$ for all $i \geq 2$. (2 points)

Attention as soft lookup

Attention: match query q to keys k1, k2, …, k5 to get weights between 0 and 1. Sum up values corresponding to each key with respective weight



Lookup: find query in database, return value corresponding to its key



Slide adapted from CS224n by Chris Manning

Self-attention/Transformer animations

- 3blue1brown animated visualization on Transformer's attention mechanism: https://www.3blue1brown.com/?topic=neural-networks&lesson=attention

- If interested, continue to watch the next chapter on MLP & how facts might be stored in a model: https://www.3blue1brown.com/?topic=neural-networks&lesson=mlp