به نام خدا

# تمرین سری پنجم درس مبانی یادگیری عمیق دکتر مرضیه داوود آبادی

# فرزان رحمانی 99521271

## سوال اول

1. (ب) و (ج) صحیح هستند.   
   در واقع معماری many-to-one RNN برای وظایف دسته بندی احساسات و تشخیص جنسیت از گفتار مناسب است. ولی برای تشخیص گفتار مناسب نیست. چرا که آن‌ها برای وظایفی طراحی شده‌اند که در آنها دنباله‌ای از ورودی‌ها داریم و باید یک خروجی واحد تولید کنیم. همان طور که میبینیم مورد (ب) و (ج) یک خروجی واحد دارند ولی مورد (آ) شامل چندین خروجی است و نیاز به معماری many-to-mant RNN دارد.

Here's why a many-to-one RNN architecture is suitable for these tasks:

Key points about many-to-one RNNs:

* They are designed for tasks where you have a sequence of inputs and need to produce a single output.
* They process the input sequence sequentially, with each element informing the next, allowing them to capture temporal dependencies within the data.

How they align with the chosen tasks:

1. Sentiment classification:
   * The input is a sequence of words (the text).
   * The output is a single sentiment label (positive or negative).
   * The RNN can capture how the sentiment unfolds over the course of the text, considering the order of words and their relationships.
2. Gender recognition from speech:
   * The input is a sequence of audio frames (the speech clip).
   * The output is a single gender label (male or female).
   * The RNN can track how vocal characteristics like pitch and intonation patterns evolve over time, which can be indicative of gender.

Why speech recognition is not a fit:

* Speech recognition requires mapping a sequence of audio frames to a sequence of text words, which necessitates a many-to-many RNN architecture that generates multiple outputs.

In essence, many-to-one RNNs excel in tasks where understanding the overall meaning or pattern within a sequence is crucial for producing a single, accurate classification.

1. گزینه (ج) صحیح است.   
   (يك طرفه، زيرا مقدار yt تنها به x1, ..., xt وابسته است و به xt+1, ..., x30 وابسته نيست.)  
   چرا که اخلاق پنبه به آب و هوای فعلی (xt) و چند روز گذشته (x1, ..., xt-1) بستگی دارد ولی به آب و هوای آینده (xt+1, ..., x30) که هنوز اتفاق نیافته است وابستگی ندارد.

Here's a breakdown of why this is the most suitable choice:

Key considerations for choosing the RNN architecture:

* Nature of the dependence: The cat's mood on a given day (y<t>) depends on the current and past weather, not on future weather. This means we only need to process the sequence in the forward direction.
* Unidirectional RNNs:
  + Process inputs sequentially from start to end.
  + Capture temporal dependencies in the forward direction.
  + Suitable for tasks where the output at any time step depends only on past and current inputs.
* Bidirectional RNNs:
  + Process inputs in both forward and backward directions.
  + Capture dependencies in both directions.
  + Best for tasks where future inputs also influence the output at a given time step (e.g., understanding a sentence where context from later words can affect the meaning of earlier words).

In this specific case:

* The cat’s mood on day t cannot be influenced by weather on days t+1, t+2, etc., as those days haven't happened yet.
* A unidirectional RNN, processing the weather sequence chronologically, can effectively model this relationship.
* Using a bidirectional RNN would be unnecessary and potentially introduce noise by considering irrelevant future information.

Therefore, a unidirectional RNN is the most appropriate choice for this problem, accurately capturing the cat's mood patterns based on the relevant weather history.

1. گزینه (ج) صحیح است. *P*(*yt|y*1*, y*2*, ..., yt-*1)   
   در یک مدل زبان ما سعی می کنیم گام بعدی را بر اساس دانش تمام مراحل قبلی پیش بینی کنیم.

Here's a breakdown of why this is the case:

Key concepts:

* Language modeling: The RNN's goal is to predict the probability of the next word in a sequence given the previous words.
* Conditional probability: The RNN estimates how likely a word is to occur based on the context of the words that have already appeared.

At the tth time step:

* The RNN has processed the words y, y<1>, ..., y<t-1>.
* It holds this information in its hidden state.
* It uses this context to estimate the probability of the next word, y<t>.

Specifically:

* P(y<t>∣y, y<1>,…,y<t−1>) represents the conditional probability of y<t> given the previous words up to y<t-1>.
* This aligns with the RNN's task of predicting the next word based on the current context.

Discarding other options:

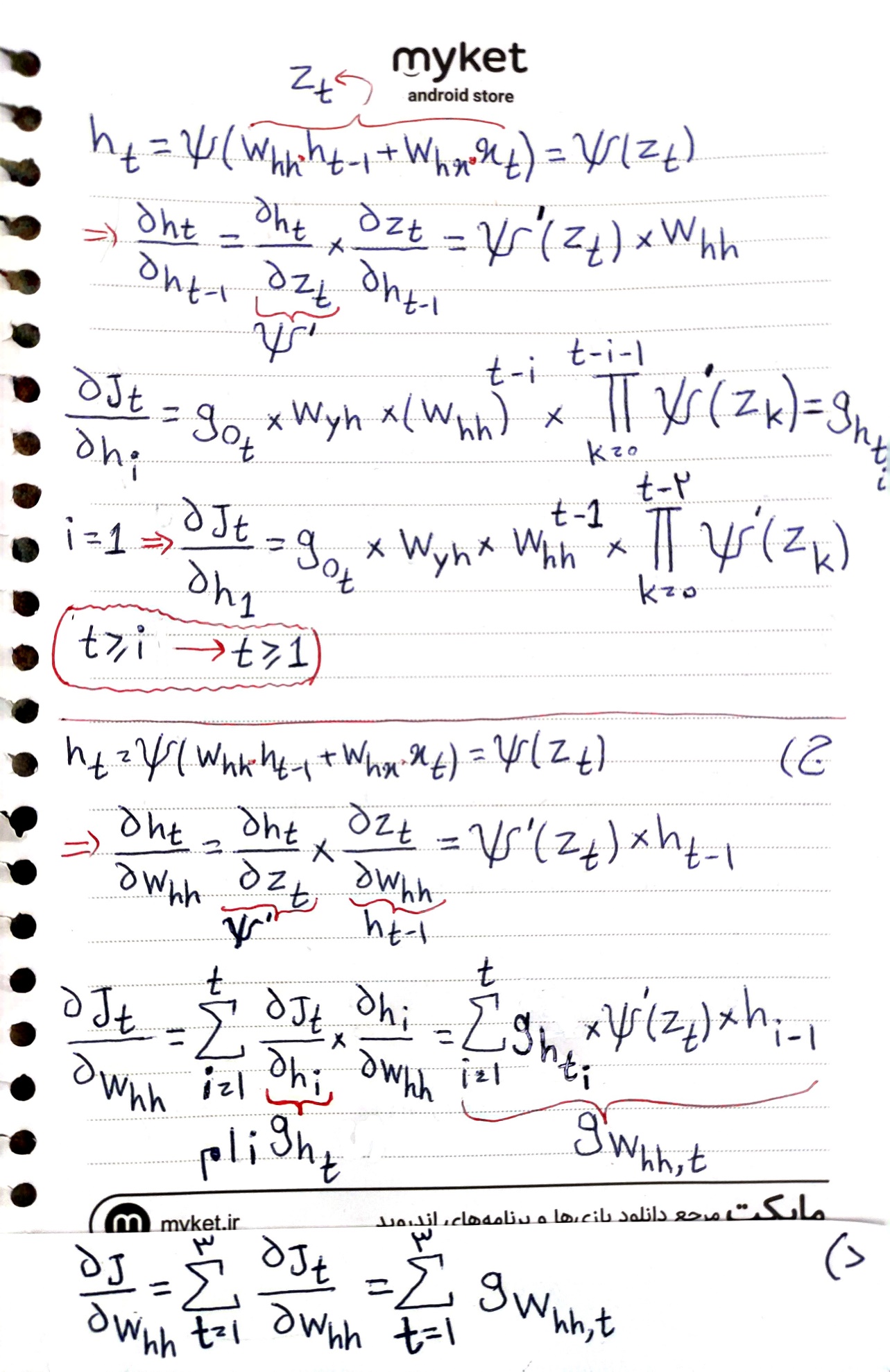
* Estimating P(y, y<1>,…,y<t−1>) would involve predicting the entire sequence up to t-1, which is not the RNN's focus at time step t.
* Estimating P(y<t>) would ignore the context of previous words, which is crucial for language modeling.
* Estimating P(y<t>∣y, y<1>,…,y<t>) is not possible as the RNN hasn't seen y<t> yet at time step t.

In conclusion, at the tth time step, the RNN is specifically estimating the conditional probability of the next word given the previous words, effectively capturing the language's sequential nature.

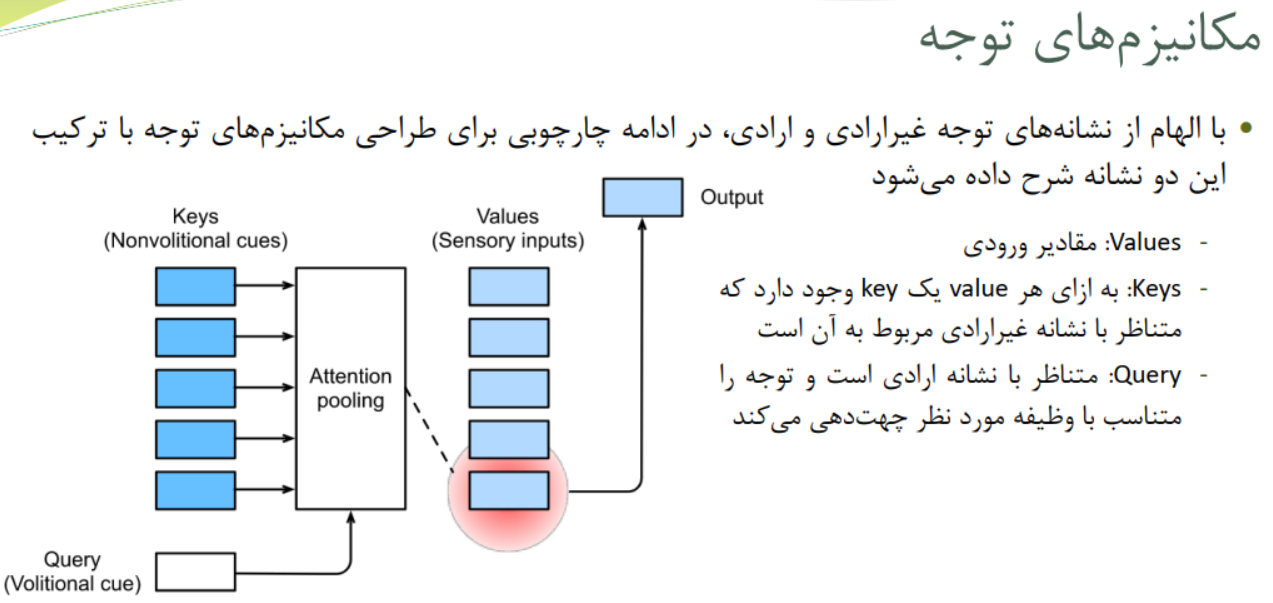
مراجع:

<https://chat.openai.com/>   
<https://bard.google.com/>   
<https://claude.ai/chats>   
<https://marcossilva.github.io/en/2019/08/12/coursera-deep-learning-module-5-week-1.html>

## سوال دوم



## سوال سوم



الف)   
مکانیسم توجه argmax مقدار کلیدی را انتخاب می‌کند که بیشترین شباهت را به پرس و جو با استفاده از ضرب داخلی دارد.  
ابتدا ضرب داخلی بین کلید ها و پرس و جو را محاسبه میکنیم تا شباهت ها را بسنجیم. سپس اندیس متناظر با بیشترین شباهت را پیدا میکنیم. در نهایت با استفاده از آن اندیس خروجی متناظر با بیشترین شباهت را به عنوان خروجی در نظر می گیریم.

در اینجا محاسبه گام به گام را میبینیم:

* 1. محاسبه ضرب داخلی بین پرس و جو و کلیدها:
* q · Keys[0] = 3 \* 1 + -1 \* 2 + -1 \* 3 = -2
* q · Keys[1] = 3 \* 2 + -1 \* 2 + -1 \* 1 = 3
* q · Keys[2] = 3 \* 0 + -1 \* 1 + -1 \* -1 = 0
* q · Keys[3] = 3 \* 0 + -1 \* -2 + -1 \* -4 = 6
  1. برای انتخاب اندیس مشابه ترین کلید، argmax را اعمال می کنیم:
* argmax([-2, 3, 0, 6]) = 3
  1. مقدار مربوط به اندیس انتخاب شده را بازیابی می کنیم:
* Output = Values[3] = [6, 1, 2]

بنابراین، خروجی لایه توجه برای پرس و جو q = [3, -1, -1] با استفاده از توجه argmax [6, 1, 2] خواهد بود.

ب)

مکانیسم توجه argmax، که در آن خروجی صرفاً توسط مشابه ترین کلید تعیین می شود، پیامدهای خاصی برای آموزش مدل های یادگیری عمیق شامل توجه دارد:

1. عدم مشتق پذیری(Non-differentiability): توجه argmax مشتق ناپذیر است. در گذر به عقب(backward pass) در طول آموزش، محاسبه گرادیان ها با توجه به کلیدها، مقادیر یا پرس و جوها چالش برانگیز می شود. این مانع از توانایی بهینه سازی مستقیم این مؤلفه ها از طریق روش های مبتنی بر گرادیان مانند پس انتشار می شود.
2. ناتوانی در یادگیری تبدیل های کلید یا پرس و جو(Inability to learn key or query transformations): از آنجایی که خروجی صرفاً بر اساس شبیه ترین کلید بدون هیچ گونه تبدیل قابل یادگیری است، مدل نمی تواند در طول فرآیند آموزش بهبود کوئری ها یا کلیدها را بیاموزد. در مقابل، مکانیسم‌هایی مانند وزن‌های توجه قابل یادگیری (به عنوان مثال، توجه مبتنی بر softmax) به شبکه اجازه می‌دهد تا وزن‌های کلیدهای مختلف را یاد بگیرد و تعمیم و یادگیری بهتری را ممکن می‌سازد.
3. Limited expressiveness: توجه argmax ممکن است برای کارهای خاصی که نیاز به درک روابط پیچیده بین کلیدها و پرس و جوها دارند، بیش از حد ساده باشد. تفاوت‌های ظریف یا شباهت در توزیع‌ها را در بر نمی‌گیرد. به سادگی مشابه ترین کلید را انتخاب می کند و احتمالاً سایر اطلاعات مرتبط را نادیده می گیرد.

به طور خلاصه، در حالی که توجه argmax ساده و از نظر محاسباتی کارآمد است، ناتوانی آن در انتشار گرادیان ها و یادگیری تبدیل ها، مفید بودن آن را در مقایسه با مکانیسم های توجه انعطاف پذیرتر و آموزش پذیرتر مانند توجه مبتنی بر softmax (به عنوان مثال، در ترانسفورماتور) محدود می کند.  
در واقع، استفاده از توجه argmax، آموزش مدل هایی که توجه را به طور موثر در بر می گیرند، دشوار می کند. در حین پس انتشار، گرادیان برای همه کلیدها به جز کلید انتخاب شده توسط argmax صفر خواهد بود. این بدان معناست که کلیدها و پرس و جوها را نمی توان بر اساس گرادیان به روز کرد تا توجه بهبود یابد، زیرا فقط یک کلید گرادیان غیر صفر می گیرد. فقدان گرادیان در اکثر کلیدها و پرس و جوها به شدت توانایی ما را برای آموزش مدل برای بهبود کیفیت توجه محدود می کند. توجه Softmax بهتر است زیرا به شیب ها اجازه می دهد تا به همه کلیدها / پرس و جوها جریان پیدا کنند و بهینه سازی توجه را در طول آموزش ممکن می کند.

همچنین جواب این قسمت به بیان bard:

Using argmax instead of softmax significantly hinders model training:

* Non-differentiable: Argmax is a discrete operation, meaning it doesn't have a gradient. This prevents backpropagation of errors through the attention layer during training.
* No gradient flow: Without gradients, the model cannot learn to adjust its queries and keys to improve attention scores, thus limiting its ability to learn meaningful attention patterns.
* One-hot attention: Argmax forces the model to focus solely on the most similar key, ignoring potentially relevant information from other keys.

In contrast, softmax offers advantages for training:

* Differentiable: Softmax produces a smooth probability distribution over keys, allowing gradients to flow through the attention layer.
* Gradient-based learning: The model can learn to refine queries and keys based on feedback from downstream tasks.
* Weighted attention: Softmax enables the model to attend to multiple keys with varying degrees of importance, leading to more nuanced and contextualized representations.

While argmax might seem conceptually simpler, its non-differentiable nature makes it unsuitable for training attention models effectively. Softmax, despite its added complexity, is crucial for enabling gradient-based learning and capturing richer attention dynamics.

In other words, Using argmax instead of softmax in attention significantly limits the model's ability to learn and adapt during training. With argmax, the attention mechanism becomes non-differentiable, making it challenging to backpropagate gradients through the network during training. This limitation prevents the model from learning to improve queries or keys based on the training data, as there are no gradients to update these parameters.  
In essence, argmax attention makes the network non-trainable with respect to attention weights. The inability to update the attention weights based on the training data results in the model's incapability to effectively capture complex patterns or relationships within the input data, reducing its overall learning capability and performance.

مراجع:

<https://chat.openai.com/>   
<https://bard.google.com/>   
<https://claude.ai/chats>

## سوال چهارم

کد این سوال در فایل DL\_HW5\_Question4.ipynb پیوست شده است.

مراجع:

<https://chat.openai.com/>   
<https://bard.google.com/>   
<https://claude.ai/chats>

پایان