به نام خدا

تمرین شماره یک درس پردازش زبانهای طبیعی — توحید عابدینی ۹۹۷۲۲۱۵۳

1-Spell Correction

سوال ۱)

کلمه ی متناظر با شماره دانشجویی: terrain

جدول ۱ برای کلمه ی پیشنهادی terrain

n	6	5	4	3	4	3	2	1
i	5	4	3	2	3	2	1	2
a	4	3	2	1	2	1	2	3
r	3	2	1	0	1	2	3	4
e	2	1	0	1	2	3	4	5
t	1	0	1	2	3	4	5	6
#	0	1	2	3	4	5	6	7
	#	**t**	**e**	**r**	**r**	**a**	**i**	**n**

جدول ۲ برای کلمه پیشنهادی terran

n	6	5	4	3	4	3	2
i	5	4	3	2	3	2	3
a	4	3	2	1	2	1	2
r	3	2	1	0	1	2	3
e	2	1	0	1	2	3	4
t	1	0	1	2	3	4	5
#	0	1	2	3	4	5	6
	#	**t**	**e**	**r**	**r**	**a**	**n**

جدول ۳ برای کلمه پیشنهادی trian

n	6	5	4	3	2	1
i	5	4	3	2	1	2
a	4	3	2	1	2	3
r	3	2	1	2	3	4
e	2	1	2	3	4	5
t	1	0	1	2	3	4
#	0	1	2	3	4	5
	#	**t**	**r**	**a**	**i**	**n**

مطابق با فاصله لونشتاین میبینیم که انتخاب های اول و سوم دارای هزینه کمتر (۱) بوده و هر دو میتوانند بهترین انتخاب باشند. یعنی terrain و train بهترین انتخاب برای terian هستند.

سوال ۲)

رشته اول متناظر با شماره دانشجویی: شماره ۳

ACTGAT

رشته دوم انتخابی: شماره ۹

AAATGCTC

	#	A	A	A	T	G	C	T	C
#	0	-2	-4	-6	-8	-10	-12	-14	-16
A	-2	↖ 1	-1	-3	-5	-7	-9	-11	-13
C	-4	-1	↖ 0	← -2	-4	-6	-6	-8	-10
T	-6	-3	-2	-1	↖ -1	-3	-5	-5	-7
G	-8	-5	-4	-3	-2	↖ 0	-2	-4	-6
A	-10	-7	-4	-3	-4	-2	↖ -1	-3	-5
T	-12	-9	-6	-5	-2	-4	-3	↖ 0	← -2

خروجی الگوریتم مچینگ زیر است که امتیاز 2- دارد.

A A A T G C T C

A C - T G A T -

سوال ۳)

به صورت کد با اسم NLP-HW02-MED.ipynb تحویل داده شده است.

2-Generative and Discriminative models

سوال ۱)

مدل مولد (Generative) برای محاسبه احتمال تعلق به یک کلاس در تسک کلاس بندی از قاعده بیز کمک میگیرد و با استفاده از likelihood و prior و marginal به محاسبه posterior از رابطه زیر میپردازد:

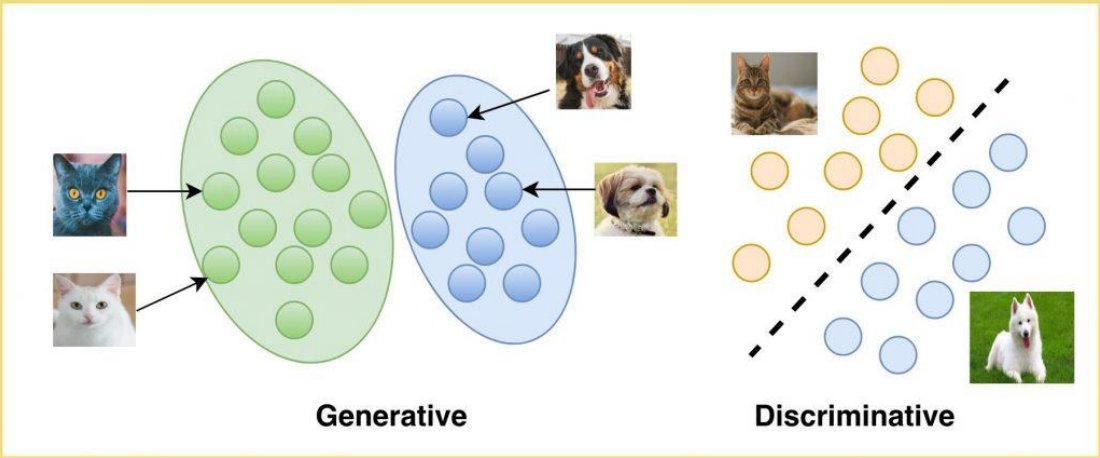
$$\underbrace{P(A|B)}_{\text{posterior}} = \underbrace{P(A)}_{\text{prior}} \times \frac{\overbrace{P(B|A)}^{\text{likelihood}}}{\underbrace{P(B)}_{\text{marginal}}}$$

و در حقیقت از دیدگاه آماری به تخمین احتمال های prior و liklehood پرداخته و با استفاده از یافتن توزیع داده های موجود اقدام به تولید (Generate) کردن داده های جدید از آن توزیع ها میکند. این مدل به این یافتن خصیصه های داده میپردازد تا با استفاده از این به کلاس بندی برسد. میتوان با نگاهی دیگر با استفاده از روابط آماری زیر این گونه تفسیر کرد که این مدل با احتمال توامان صورت رابطه ی بالا را محاسبه میکند و سپس از قاعده ی بیز به محاسبه احتمال posterior میپردازد.

$$P(A,B) = P(A \mid B)P(B) = P(B \mid A)P(A)$$

در حالی که مدل Discriminative احتمال شرطی posterior را به صورت مستقیم از داده ها محاسبه میکند. مدل Discriminative کاری با خصیصه های داده ندارد و تفکیک را از روی داده ها انجام میدهد و دانش عمقی راجع به ساختار داده ها ندارد.

شکل زیر میتواند دید بهتری راجع به تفاوتهای این دو نوع مدل به ما بدهد:



برای مثال میتوان مدل های بیز و مدل های مارکوفی و شبکه های GAN و ... را برای مدل های مولد نام برد و برای مدل های Discriminative هم میتوان به SVM و RF و DT و LR و... اشاره کرد.

منابع:

https://learnopencv.com/generative-and-discriminative-models/
https://towardsdatascience.com/introduction-to-generative-and-discriminative-models-9c9ef152b9af
https://stackoverflow.com/questions/879432/what-is-the-difference-between-a-generative-and-a-discriminative-algorithm

سوال ۲)

رده بند نایو بیز به دلیل ماهیتی که دارد و هر کلمه را جدا جدا و به صورت مستقل بررسی میکند و در این حالت کلمات دو بخشی مانند Hong Kong و San Fransisco و Islamic Republic هر بخششان به صورت جدا در محاسبه حاصل ضرب احتمالات حضور پیدا میکند و احتمال محاسبه شده با واقعیت تفاوت دارد اما رده بند MaxEnt اینگونه نیست و با استفاده از فیچر ها و همچنین وزن دهی به فیچر های مختلف این امر مدیریت میشود و تعدد ویژگی هایی که همبستگی بالا با هم داشته باشند را نداریم (بر خلاف نایو بیز) .
همچنین میتوان گفت که نایو بیز از نوع مدل های Generative است در حالی که MaxEnt یک مدل Discriminative میباشد و تمامی تفاوت هایی که برای سوال یک گفته شد برای این دو رده بند نیز صادق است.

سوال ۳)

بسته به شرایط مساله و یا ترجیحات میتوان از هر دو نوع مدل یا حداقل مدل ترکیبی برای حل این مساله استفاده کرد. راه حل های پیشنهادی عمدتا حالت ترکیبی یا Generative دارند. یک راه حل پیشنهادی برای تسک SRL (که در حقیقت یک sub-task است.) میتواند استفاده از LSTM باشد که بیشتر حالت Generative دارد. از امبدینگ های Word2vec به همراه یک امبدینگ تصادفی و یک امبدینگ POS تصادفی به صورت concat شده به عنوان امبدینگ اولیه استفاده شده است. یک بیت نشانگر هم به امبدینگ اضافه میکنیم که بیانگر مسند (predicate) بودن این کلمه است. تا شبکه با انواع مختلف مسند و غیر مسند رفتار متفاوتی داشته باشد. این امبدینگ ها به یک LSTM دوطرفه داده میشوند تا context کلمه در جمله مشخص گردد. در نهایت با ضرب نقطه ای state مخفی کلمه در LSTM و state مخفی مسند در همان شبکه و اعمال softmax کلمه مورد نظر را برچسب زنی میکنیم.

$$p(r|v_i, v_p) \propto \exp(W_r(v_i \cdot v_p))$$

رابطه زیر همانگونه که در بالا توضیح داده شد احتمال برچسب ها (role-r) ها را با توجه به vectorهای ورودی i و مسند p و softmax گرفتن با exp حساب میکند. که به دلیل مخرج ثابت یک softmax از علامت تناسب استفاده کرده که نقش مخرج فقط این است که جمع احتمالات یک شود. و همچنین ضرب نقطه ای هم در فرمول قابل مشاهده است.

$$W_{l,r} = ReLU\,(U(u_l \cdot v_r))$$

همچنین میتوان ماتریس وزن ها را برای برچسب r به صورت بالا مدل کرد که v_l همان وکتور تصادفی مرتبط با lemma و v_r همان وکتور تصادفی مرتبط با role یا نقش است که در نهایت تابع فعال ساز ReLU هم دارد.

سوال ۴)

جدول ۳:

P(a,b)	b=1	b=2	b=3
a=0	1/10	1/10	2/10
a=1	2/10	0	1/10
a=2	1/10	1/10	1/10

جدول ۴:

P(a|b)	b=1	b=2	b=3
a=0	1/4	1/2	2/4
a=1	2/4	0	1/4
a=2	1/4	1/2	1/4

منبع:

https://stackoverflow.com/questions/879432/what-is-the-difference-between-a-generative-and-a-discriminative-algorithm

---

3-Text Classification

سوال ۱)

کلمه نایو در نایو بیز به این معنا است که این کلاسیفایر کلمات مختلف ورودی را به صورت مستقل از هم بررسی میکند. یعنی ویژگی ها به صورت مستقل از هم فرض میشوند که به راحتی محاسبات و حل مساله منجر میشود. معنی نایو هم یعنی ساده (خام-بی تجربه) که بدین معنی است که ممکن است این فرض صدق کند یا نکند!

سوال ۲)

جمله پیشنهادی برای مجموعه آموزش:

I feel happy today. / Emotion:Happy

برای اینکه smoothing انجام دهیم، در ابتدا تعداد کل کلمات مجموعه داده آموزش را به دست می آوریم:

تعداد کل کلمات متن: 28 کلمه

با استفاده از روابط زیر به محاسبه احتمال ها و رده بندی باید بپردازیم:

$$\hat{P}(w \mid c) = \frac{count(w,c)+1}{count(c)+|V|}$$

$$\hat{P}(c) = \frac{N_c}{N}$$

جملات مجموعه داده تست:

d1: I feel relaxed and lucky after my job interview
d2: This movie was boring and predictable with no fun
d3: The movie was very wonderful.

به محاسبه احتمالات میپردازیم.

ابتدا احتمالات پیشین کلاس ها را محاسبه میکنیم. تعداد جملات مجموعه آموزش ما ۷ است و بنابراین داریم:

$$P(Happy) = \frac{4}{7}$$

$$P(sadness) = \frac{3}{7}$$

همچنین داریم:

$$count\ (Happy) = 21$$

$$count\ (sadness) = 18$$

در نهایت احتمالات شرطی برای تک تک کلمات موجود در جملات تست را محاسبه میکنیم.

$$P(I|Happy) = \frac{(2+1)}{(21+28)} = \frac{3}{49}$$

$$P(I|sadness) = \frac{(2+1)}{(18+28)} = \frac{3}{46}$$

$$P(feel|Happy) = \frac{(2+1)}{(21+28)} = \frac{3}{49}$$
$$P(feel|sadness) = \frac{(0+1)}{(18+28)} = \frac{1}{46}$$

$$P(relaxed|Happy) = \frac{(1+1)}{(21+28)} = \frac{2}{49}$$
$$P(relaxed|sadness) = \frac{(0+1)}{(18+28)} = \frac{1}{46}$$

$$P(and|Happy) = \frac{(1+1)}{(21+28)} = \frac{2}{49}$$
$$P(and|sadness) = \frac{(1+1)}{(18+28)} = \frac{2}{46}$$

$$P(lucky|Happy) = \frac{(1+1)}{(21+28)} = \frac{2}{49}$$
$$P(lucky|sadness) = \frac{(0+1)}{(18+28)} = \frac{1}{46}$$

$$P(after|Happy) = \frac{(0+1)}{(21+28)} = \frac{1}{49}$$
$$P(after|sadness) = \frac{(0+1)}{(18+28)} = \frac{1}{46}$$

$$P(my|Happy) = \frac{(1+1)}{(21+28)} = \frac{3}{49}$$
$$P(my|sadness) = \frac{(1+1)}{(18+28)} = \frac{3}{46}$$

$$P(job|Happy) = \frac{(0+1)}{(21+28)} = \frac{1}{49}$$
$$P(job|sadness) = \frac{(1+1)}{(18+28)} = \frac{2}{46}$$

$$P(interview|Happy) = \frac{(0+1)}{(21+28)} = \frac{1}{49}$$
$$P(interview|sadness) = \frac{(0+1)}{(18+28)} = \frac{1}{46}$$

$$P(This|Happy) = \frac{(0+1)}{(21+28)} = \frac{1}{49}$$
$$P(This|sadness) = \frac{(0+1)}{(18+28)} = \frac{1}{46}$$

$$P(movie|Happy) = \frac{(1+1)}{(21+28)} = \frac{2}{49}$$
$$P(movie|sadness) = \frac{(0+1)}{(18+28)} = \frac{1}{46}$$

$$P(was|Happy) = \frac{(0+1)}{(21+28)} = \frac{1}{49}$$
$$P(was|sadness) = \frac{(1+1)}{(18+28)} = \frac{2}{46}$$

$$P(boring|Happy) = \frac{(0+1)}{(21+28)} = \frac{1}{49}$$
$$P(boring|sadness) = \frac{(1+1)}{(18+28)} = \frac{2}{46}$$

$$P(predictable|Happy) = \frac{(0+1)}{(21+28)} = \frac{1}{49}$$
$$P(predictable|sadness) = \frac{(0+1)}{(18+28)} = \frac{1}{46}$$
$$P(with|Happy) = \frac{(0+1)}{(21+28)} = \frac{1}{49}$$
$$P(with|sadness) = \frac{(0+1)}{(18+28)} = \frac{1}{46}$$

$$P(no|Happy) = \frac{(0+1)}{(21+28)} = \frac{1}{49}$$

$$P(no|sadness) = \frac{(0+1)}{(18+28)} = \frac{1}{46}$$

$$P(fun|Happy) = \frac{(1+1)}{(21+28)} = \frac{2}{49}$$
$$P(fun|sadness) = \frac{(0+1)}{(18+28)} = \frac{1}{46}$$

$$P(The|Happy) = \frac{(0+1)}{(21+28)} = \frac{1}{49}$$
$$P(The|sadness) = \frac{(0+1)}{(18+28)} = \frac{1}{46}$$

$$P(very|Happy) = \frac{(1+1)}{(21+28)} = \frac{2}{49}$$
$$P(very|sadness) = \frac{(0+1)}{(18+28)} = \frac{1}{46}$$

$$P(wonderful|Happy) = \frac{(0+1)}{(21+28)} = \frac{1}{49}$$
$$P(wonderful|sadness) = \frac{(0+1)}{(18+28)} = \frac{1}{46}$$

حال برای محاسبه احتمال هر جمله از رابطه زیر کمک میگیریم:

$$P(d|c) \propto P(c) * P(w_1|c) * P(w_2|c) * \dots$$

برای جمله اول تست داریم:

$$\begin{aligned} P(d1|Happy) &\propto P(Happy) * P(I|Happy) * P(feel|Happy) \\ &* P(relaxed|Happy) * P(and|Happy) * P(lucky|Happy) \\ &* P(after|Happy) * P(my|Happy) * P(job|Happy) \\ &* P(interview|Happy) \\ &= \frac{4}{7} * \frac{3}{49} * \frac{3}{49} * \frac{2}{49} * \frac{2}{49} * \frac{2}{49} * \frac{1}{49} * \frac{3}{49} * \frac{1}{49} * \frac{1}{49} \\ &= 7.5796819e-14 \end{aligned}$$

$$P(d1|sadness)$$
$$\propto P(sadness) * P(I|sadness) * P(feel|sadness)$$
$$* P(relaxed|sadness) * P(and|sadness) * P(lucky|sadness)$$
$$* P(after|sadness) * P(my|sadness) * P(job|sadness)$$
$$* P(interview|sadness)$$
$$= \frac{3}{7} * \frac{3}{46} * \frac{1}{46} * \frac{1}{46} * \frac{2}{46} * \frac{1}{46} * \frac{1}{46} * \frac{3}{46} * \frac{2}{46} * \frac{1}{46}$$
$$= 1.6730358e - 14$$

جمله اول به دلیل اینکه $P(d1|Happy) > P(d1|sadness)$ است متعلق به کلاس Happy است.

برای جمله دوم تست داریم:

$$P(d2|Happy)$$
$$\propto P(Happy) * P(This|Happy) * P(movie|Happy)$$
$$* P(was|Happy) * P(boring|Happy) * P(and|Happy)$$
$$* P(predictable|Happy) * P(with|Happy) * P(no|Happy)$$
$$* P(fun|Happy) = \frac{4}{7} * \frac{1}{49} * \frac{2}{49} * \frac{1}{49} * \frac{1}{49} * \frac{2}{49} * \frac{1}{49} * \frac{1}{49} * \frac{1}{49} * \frac{2}{49}$$
$$= 2.8072896e - 15$$

$$P(d2|sadness)$$
$$\propto P(sadness) * P(This|sadness) * P(movie|sadness)$$
$$* P(was|sadness) * P(boring|sadness) * P(and|sadness)$$
$$* P(predictable|sadness) * P(with|sadness) * P(no|sadness)$$
$$* P(fun|sadness) = \frac{3}{7} * \frac{1}{46} * \frac{1}{46} * \frac{2}{46} * \frac{2}{46} * \frac{2}{46} * \frac{1}{46} * \frac{1}{46} * \frac{1}{46} * \frac{1}{46}$$
$$= 3.7178573e - 15$$

جمله دوم به دلیل اینکه $P(d2|Happy) < P(d2|sadness)$ است متعلق به کلاس sadness است.

برای جمله سوم تست داریم:

$$P(d3|Happy) \propto P(Happy) * P(The|Happy) * P(movie|Happy) * P(was|Happy) * P(very|Happy) * P(wonderful|Happy) = \frac{4}{7} * \frac{1}{49} * \frac{2}{49} * \frac{1}{49} * \frac{2}{49} * \frac{1}{49} = 8.09173297e-9$$

$$P(d3|sadness) \propto P(sadness) * P(The|sadness) * P(movie|sadness) * P(was|sadness) * P(very|sadness) * P(wonderful|sadness) = \frac{3}{7} * \frac{1}{46} * \frac{1}{46} * \frac{2}{46} * \frac{1}{46} * \frac{1}{46} = 4.16163562e-9$$

جمله سوم به دلیل اینکه $P(d3|Happy) > P(d3|sadness)$ است متعلق به کلاس Happy است.

سوال ۳) در نوت بوک با اسم Bayes_Classifier_Notebook.ipynb تحویل شده است.

4- A Warm-Up for Deep Learning

سوال ۱)

برای رده بندی چند کلاسه از تابع فعال ساز softmax استفاده میشود. به این دلیل که خروجی ها باید حالت احتمالی داشته باشند.یعنی نا منفی باشند که این شرط را دارد و همچنین جمع خروجی ها هم باید یک شود که توجه به فرمول این تابع فعال ساز برابر با یک خواهد شد که یک نگاشت وزن دار از خروجی نورون ها است. وقتی دو کلاس داریم میتوان از سیگموید استفاده کرد اما وقتی با بیش از دو کلاس مواجهیم باید از softmax استفاده کنیم.

رابطه تابع فعال ساز softmax:

$$Softmax(Z_i) = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \text{ for i=1 to k}$$

Softmax equation

سوال ۲)

CUDA برای موازی سازی کد ها و محاسبات در یادگیری عمیق استفاده میشود. ماهیت محاسبات در شبکه های عصبی بیشتر حالت ضرب ماتریسی دارند که یک عملیات ذاتا موازی و غیر وابسته به بقیه عملیات ها است. پس میتوان با موازی سازی زمان محاسبات را به میزان چشمگیری کاهش داد. میدانیم کارتهای گرافیکی تعداد هسته بیشتری (والبته هر هسته ضعیف تر است چون قرار است محاسبات ساده اما پر تعداد را انجام دهد) از CPU دارند و برای کارهای موازی بهتر هستند. برای اینکه کد نوشته شده بتواند بر روی کارت گرافیکی اجرا شود از CUDA استفاده میشود که امکان اجرا شدن یک کد را روی کارت گرافیکی برای ما فراهم میکند.

سوال ۳)

انسان ها به صورت شناختی در حال انجام کار ها توجه خود را متمرکز به بخش خاصی میکنند و تا حدی جزییات دیگر را در نظر نمیگیرند و کمتر به آنها توجه میکنند. ایده ی اصلی مکانیسم توجه در یادگیری عمیق از این ویژگی شناختی انسان گرفته شده است که به وجود آمدن ترنسفورمور ها انجامید که تحول بزرگی در یادگیری عمیق داشت.

منبع:

https://www.analyticsvidhya.com/blog/2019/11/comprehensive-guide-attention-mechanism-deep-learning/

سوال ۴)

دلایل و محدودیت های موجود در هوش مصنوعی نمادین باعث شد که این رویکرد تا حدی جای خود را به رویکرد اتصال محور (Connectionist) بدهد که با تقلید از عملکرد مغز انسان به حل مسائل میپردازد. هوش نمادین در مسائلی که محدود و خاص هستند عملکرد خوبی دارد اما قدرت تعمیم ضعیف تری دارد و در شرایطی که دانش کافی از محیط و مساله نداریم تا قوانین مناسب استخراج کنیم به مشکل میخوریم. هوش نمادین در برابر تغییرات آسیب پذیر است و ظهور گونه ها یا نوع موجودیت یا هر مفهوم جدید در مساله مشکل زا خواهد شد و همین مساله یک موضوع سوال برانگیز است و مفهوم آموزش را در این نوع رویکرد زیر سوال

میبرد که آیا صرفا یک استدلال سطحی است یا آموزشی هم در میان است؟! در حقیقت هوش نمادین خودش توانایی یادگیری ندارد.
مساله ی اصلی موجود که منجر به بازنگری در استفاده از هوش نمادین در برخی زمینه ها شد مساله ی common sense knowledge problem است که وجود یک پایگاه داده ی جامع برای دانش کلی و جامع از تمامی حس های مشترک را ایجاب میکند که در عمل چنین چیزی ناممکن است. چون به تعداد بسیار زیاد گزاره های دانشی و استدلالی ضمنی داریم که توانایی تجمیع آنها کنار یک دیگر محل سوال است.

منبع:

https://iep.utm.edu/connect/
https://becominghuman.ai/symbolic-ai-vs-connectionism-9f574d4f321f
https://www.techslang.com/what-is-symbolic-ai-examining-its-successes-and-failures/

سوال ۵)

بخش a) Tensor: همانطوری که بالا تر اشاره شد بیشتر کار ما با ماتریس ها و ضرب ماتریسی است. پس باید به دنبال یک ساختار داده خاص در این زمینه بود. به طور کلی تنسور یک تعمیم برای نوع داده ماتریس و آرایه است که به در آن داده ها به صورت منظم و روی یک grid یا شبکه مربعی پخش شده اند. سعی ما بر این است که تمامی داده ها را به صورت تنسور در بیاوریم و عملیات جبری را روی آن انجام دهیم.
تنسور ها هر ابعادی میتوانند داشته باشند. شکل زیر یک نمونه تنسور ۳ در ۳ در ۳ (۳ بعدی) است که برای درک بهتر آورده شده است.

```
(3, 3, 3)
[[[ 1  2  3]
  [ 4  5  6]
  [ 7  8  9]]

 [[11 12 13]
  [14 15 16]
  [17 18 19]]

 [[21 22 23]
  [24 25 26]
  [27 28 29]]]
```

بخش b) Embedding: امبدینگ یک بازنمایی عددی از فضای اصلی ماست که ابعاد کمتری در مقایسه با فضای اصلی دارد و طی یک فرآیند یادگیری ایجاد میشود و در قالب وکتور برای هر کلمه ذخیره میشود. این بازنمایی عددی در صورتی که برخی خصیصه ها را داشته باشد بهتر خواهد بود و کاربرد بیشتری خواهد داشت.

به عنوان مثال اگر تفاوت و معنای کلمات را هم بتواند خوب نمایش دهد و از نظر حافظه و حجم هم بهینه باشد بسیار کاربردی تر خواهد بود.
شکل زیر یک نمونه از امبدینگ های ایجاد شده برای لغات زبان انگلیسی در کتابخانه پایتورچ با لایه Embedding است:

		1	2	3	4	5	6	7	.....	
1	Apple	0.9898	0.7865	0.5645	0.7509	0.4534	0.5467	0.6498	0.7613	0.8
2	Banana	0.4533	0.8644	0.1538	0.4313	0.3511	0.2422	0.2422	0.3553	0.2
3	Cat	0.8734	0.8363	0.4821	0.1378	0.2341	0.2122	0.6775	0.3432	0.
4	Dog	0.9873	0.4836	0.1342	0.19564	0.2131	0.3433	0.2244	0.7453	0.5
5	Eag	0.9473	0.4836	0.4343	0.9211	0.1221	0.4634	0.7464	0.2424	0.5
6	Google	0.7634	0.4836	0.1313	0.1344	0.1232	0.6222	0.6564	0.3522	0.3
7	Home	0.8463	0.9732	0.4411	0.1333	0.6453	0.3435	0.3535	0.2442	0.3
.....	.	0.8653	0.4835	0.1343	0.4421	0.7567	0.2424	0.5241	0.3221	0.3
000	Zoo	0.4736	0.9473	0.1453	0.1134	0.6564	0.1749	0.1892	0.1344	0.3

Word2Vec , Glove و... چند نمونه از امبدینگ های معروف در زمینه پردازش متن هستند که به ذکر نام آنها بسنده میشود.

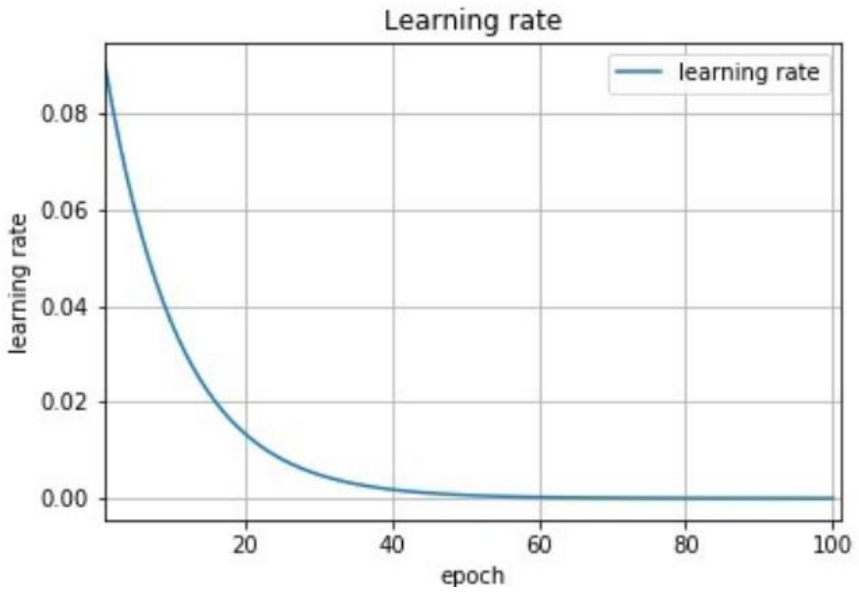
بخش c) Representation: در یادگیری ماشین و هوش مصنوعی که به طور کلی ما با کامپیوتر ها سر و کار داریم و تنها چیز قابل درک برای آنها اعداد است. پس ما باید تمامی اطلاعات خود را در قالب اعداد باز نمایی کنیم. به طور کلی به مجموعه ویژگی هایی که ما به کمک آنها یک نمونه را مدل میکنیم یک بازنمایی با Representation از آن میگویند. برای مثال در یک تصویر ۳۲*۳۲ پیکسل که به صورت RGB است یک تنسور سه بعدی ۳۲در ۳۲ در ۳ که تمامی پیکسل های این تصویر را در خود دارد که هر یک یک عدد صحیح بین ۰ تا ۲۵۵ (در اسکیل ۸ بیتی) هستند یک بردار ویژگی میگوییم که خود یک باز نمایی است. در پردازش متن هم میتوان به مجموعه بردار های امبدینگ تک تک کلمات یک جمله در کنار هم یک بازنمایی از آن جمله گفت.
بخش d) Optimizer: ما در یادگیری ماشین در حال بهینه کردن یک مساله هستیم. برای کمینه کردن یک تابع خطا باید یک مساله ی بهینه سازی را حل کنیم. شبکه های عصبی و یا هر الگوریتم دیگری که داشته باشیم دارای یک سری پارامتر است و در نهایت برای ما یک خروجی میدهد که این خروجی را توسط یک تابع هزینه ارزیابی میکنیم. به عمل مینیمم کردن این تابع هزینه optimize کردن و به الگوریتمی که بتواند این را انجام دهد optimizer گفته میشود
گرادیان گیری و حرکت در جهت کاهش گرادیان پایه و اساس بسیاری (یا حتی شاید بتوان گفت تمامی) بهینه ساز ها (optimizer) ها است. برای چند نمونه میتوان به stochastic gradient dscent , adam, rmsprop به عنوان چند بهینه ساز اشاره کرد.

بخش e) Scheduler: تنظیم یک نرخ یادگیری مناسب در طول عمل بهینه سازی یک چالش بزرگ و مساله مهم در یادگیری عمیق است. یک سری الگوریتم های scheduler برای برنامه ریزی نحوه تعییر نرخ یادگیری در این فرآیند ایجاد شده است که این کار را انجام میدهند. به طور غالب ما نیاز به نرخ یادگیری بزرگ در ابتدای فرآیند و نرخ یادگیری کوچیک تر گام های انتهایی داریم.

یک مثال میتواند کاهش نمایی نرخ یادگیری باشد. که با رابطه ی زیر حساب میشود:

$$\alpha = (decayRate^{epochNumber}) * \alpha 0$$

یک نرخ کاهش داریم که به عنوان مثال ۰.۹۹ است که به توان تعداد ایپاک ها رسیده و در میزان نرخ یادگیری قبلی ضرب میشود تا کمی آن را کاهش دهد. میتوان مقدار پارامتر نرخ کاهش را تغییر داد تا سرعت را کنترل کرد. شکل زیر نحوه کاهش نرخ یادگیری را در این الگوریتم برای یک کانفیگ خاص نشان میدهد:



که البته صرفا یک تصویر برای درک مفهوم است و راجع به عملکرد خوب این کانفیگ و اینکه نرخ یادگیری با این سرعت افت کند و در نهای به صفر برسد نمیتوان نظر قطعی داد و باید در مساله بررسی شود. مثلا میتوان این کاهش را هر چند ایپاک یک بار اعمال کرد.

منبع:

https://machinelearningmastery.com/introduction-to-tensors-for-machine-learning/
https://machinelearningmastery.com/what-are-word-embeddings/
https://androidkt.com/pre-train-word-embedding-in-pytorch/
https://medium.com/mlearning-ai/optimizers-in-deep-learning-7bf81fed78a0
https://towardsdatascience.com/learning-rate-scheduler-d8a55747dd90
https://towardsdatascience.com/learning-rate-schedules-and-adaptive-learning-rate-methods-for-deep-learning-2c8f433990d1