



دانشگاه تربیت مدرس

دانشکده علوم ریاضی

پایان نامه دوره کارشناسی ارشد علوم کامپیوتر

روش های عمیق مبتنی بر مبدل های بینایی در  
تحلیل داده های تصویری

توسط

سید محمد بادزهره

استاد راهنما

آقای دکتر منصور رزقی

پاییز ۱۴۰۳

تقدیم به

پدر بزرگوار و مادر مهربانم و برادر عزیزم  
آن‌ها که از خواسته‌هایشان گذشتند، سختی‌ها را به جان خریدند و خود را سپر بلای مشکلات و  
ناملایمات کردند تا من به جایگاهی که اکنون در آن ایستاده‌ام برسم.

## قدردانی

از استاد کرامت‌دور، جناب آقای دکتر رزقی که بارها بنامی‌های دلسوزانه و ارزشمند خود، همواره در مسیر تحقیق این پایان‌نامه یار و راهنمای من بودند، نهایت سپاس و قدردانی را دارم.

از خانواده عزیزم که با محبت بی‌پایان، صبوری و حمایت‌های بی‌دریغ‌شان، همواره پشتیبان من در طی این مسیر سخت و پرچالش بودند، صمیمانه سپاسگزارم.

سید محمد باذخره

پاییز ۱۴۰۳

## چکیدہ

بيعلعذدقفللعقفلدخقفدللقفللقفا

# فهرست مطالب

ج	فهرست جداول
د	فهرست تصاویر
۱	پیش‌گفتار
۲	۱ مفاهیم اولیه
۲	۱.۱ مقدمه
۲	۱.۱.۱ آغاز هوش مصنوعی و هدف اصلی
۳	۲.۱.۱ دوره طلایی و پیشرفت‌های اولیه
۳	۳.۱.۱ انتظارات بیش از حد و ظهور عصر تاریک
۳	۴.۱.۱ عوامل اصلی عصر تاریک هوش مصنوعی
۴	۵.۱.۱ پایان عصر تاریک و بازگشت هوش مصنوعی
۵	۲.۱ انواع مدل یادگیری ماشین و شبکه‌های عصبی
۵	۱.۲.۱ یادگیری ماشین: مروری کلی
۵	۲.۲.۱ تقسیم‌بندی‌های اصلی در یادگیری ماشین
۵	۳.۲.۱ یادگیری نظارت شده (Supervised-Learning)
۷	۴.۲.۱ معرفی چند مدل از الگوریتم یادگیری کلاسیک

۵.۲.۱	ماشین بردار پشتیبان: (Support-Vector-Machine)	۸
۶.۲.۱	بیز ساده: (Naive-Bayes)	۸
۷.۲.۱	معایب:	۹
۸.۲.۱	شبکه‌های عصبی بازگشتی (RNN) و شبکه‌های حافظه بلند مدت	
	کوتاه مدت (LSTM)	۹

## فهرست جداول

## فهرست تصاویر



## پیش گفتار

قدشتمقدکنقصدبثقلدقفخدلqxفادخفادخ

# فصل ۱

## مفاهیم اولیه

در این فصل به معرفی مقدمات و مفاهیم مورد نیاز در این پایان نامه می پردازیم.

### ۱.۱ مقدمه

در این بخش به تاریخچه هوش مصنوعی، دستاورد های اولیه، چالش ها، دلایل رکود هوش مصنوعی و پایان عصر تاریک هوش مصنوعی صحبت میکنیم

#### ۱.۱.۱ آغاز هوش مصنوعی و هدف اصلی

هوش مصنوعی به عنوان شاخه ای از علوم کامپیوتر، در دهه ۱۹۵۰ با هدف ساخت سیستم ها و ماشین هایی که توانایی تقلد از هوش انسانی را دارند، آغاز شد. نخستین بار مکاری در سال ۱۹۵۶ این اصطلاح را به کار گرفت. و هوش مصنوعی به عنوان علمی که در آن به مطالعه الگوریتم هایی برای تقلید رفتار انسانی می پردازد، شناخته شد. اهداف اولیه هوش مصنوعی شامل توانایی درک زبان، یادگیری، حل مسئله و تولید موجودات هوشمند بود. در این دوران پروژه های تحقیقاتی زیادی

به امید دستیابی به هوش مصنوعی عمومی (Artificial AGI) General (Intelligence) شروع به کار کردند

### ۲.۱.۱ دوره طلایی و پیشرفت‌های اولیه

در دهه ۵۰ و ۶۰ میلادی، هوش مصنوعی به عنوان یکی از پرچمداران پژوهش‌های نوین شناخته می‌شد. الگوریتم‌های اولیه به کمک روش‌های منطقی و ریاضیاتی برای حل مسئله و بازی‌های ساده توسعه یافتند مانند انواع الگوریتم جستجوی درختی که در این دوره به وجود آمدند و زمینه ساز اولین دستاوردهای هوش مصنوعی در بازی‌های تخته‌ای همچون شطرنج شدند. در این دوران پیشرفت‌های بیشتری در پردازش زبان طبیعی (NLP) و سیستم‌های خبره (Expert Systems) نیز صورت گرفت که این امید را در دانشمندان و محققان تقویت کرد که دستیابی به هوش مصنوعی عمومی به زودی ممکن خواهد بود.

### ۳.۱.۱ انتظارات بیش از حد و ظهور عصر تاریک

با وجود پیشرفت‌های هوش مصنوعی، محدودیت‌های تکنولوژی مثل gpu ها و محاسباتی در آن زمان و همچنین کمبود داده‌های کافی برای آموزش مدل‌های پیچیده‌تر، باعث شد که بسیاری از پروژه‌های تحقیقاتی نتوانند به نتایج پیش‌بینی شده دست یابند. و در نتیجه، هوش مصنوعی در دهه ۷۰ به مرحله‌ای از رکود وارد شد که به آن عصر تاریک هوش مصنوعی یا (AI Winter) می‌گویند. در این دوران بسیاری از پروژه‌ها تعطیل و سرمایه‌گذاری‌ها قطع شدند و دولت‌ها و سازمان‌های سرمایه‌گذار به دلیل عدم دستیابی به نتایج مطلوب از ادامه سرمایه‌گذاری منصرف شدند.

### ۴.۱.۱ عوامل اصلی عصر تاریک هوش مصنوعی

- محدودیت‌های سخت‌افزاری: در آن زمان، سیستم‌های اولیه هوش مصنوعی به محاسبات سنگینی نیاز داشتند که با توان پردازشی محدود آن زمان همخوانی نداشت.

● کمبود داده‌ها: در آن زمان، دسترسی به داده‌های کافی برای آموزش مدل‌های پیچیده ممکن نبود و الگوریتم‌های موجود به داده‌های بیشتری نیاز داشتند تا بتوانند به درستی آموزش ببینند و عملکرد مطلوبی داشته باشند.

● روش‌های محدود یادگیری: الگوریتم‌های اولیه به شدت به برنامه‌ریزی انسانی وابسته بودند و در بسیاری از موارد، مدل‌ها قادر به تعمیم به مسائل جدید نبودند و نمی‌توانستند تعمیم‌پذیری خیلی بالایی داشته باشند. [۲].

### ۵.۱.۱ پایان عصر تاریک و بازگشت هوش مصنوعی

پس از چندین سال رکود و عدم سرمایه‌گذاری در حوزه هوش مصنوعی، سرانجام در دهه ۱۹۸۰ و ۱۹۹۰ عصر تاریک هوش مصنوعی با تحولات تکنولوژی و از همه مهم‌تر ظهور سیستم‌های خبره (Expert-Systems) به پایان رسید. سیستم‌های خبره به عنوان یکی از اولین تلاش‌های موفق برای کاربردهای صنعتی در هوش مصنوعی به وجود آمدند. برخلاف الگوریتم‌های اولیه، این سیستم‌ها از پایگاه بزرگ قواعد و قوانین (Rule-Based-Systems) استفاده می‌کردند. در سیستم‌های خبره به جای تلاش برای شبیه‌سازی کلی هوش مصنوعی، بر حل مسائل تخصصی برای صنایع و سازمان‌ها تمرکز می‌کردند. برای مثال، سیستم‌های خبره در پزشکی برای تشخیص بیماری‌ها و پیشنهاد درمان، در صنعت برای مدیریت و پیش‌بینی خرابی ماشین‌آلات، و در امور مالی برای تحلیل و ارزیابی ریسک کاربرد داشتند. هرچند این سیستم‌ها نمی‌توانستند درک عمیق و هوشمندی عمومی را ایجاد کنند. اما برای رفع نیازهای پیچیده مناسب بودند. همزمان با موفقیت این سیستم‌ها، بهبودهای زیادی در سخت‌افزارها و کاهش هزینه‌های پردازش به وجود آمد. در دهه‌های ۱۹۸۰ و ۱۹۹۰ کامپیوترها به تدریج قوی‌تر و مقرون به‌صرفه‌تر شدند و امکان پردازش داده‌های بیشتر و اجرای الگوریتم‌های پیچیده‌تر فراهم شد. این افزایش توان محاسباتی، نیاز به پردازش داده‌های بزرگ و پیچیده را فراهم کرد و در نتیجه دسترسی به داده‌ها و انجام محاسبات سنگین برای توسعه الگوریتم‌های جدید فراهم شد. از طرف دیگر، پیشرفت‌های انجام شده در ذخیره‌سازی داده

و رشد اینترنت باعث دسترسی گسترده تر به داده ها و منابع اطلاعاتی شد. به این ترتیب، مجموعه ای از عوامل شامل ظهور سیستم های خبره، افزایش قدرت پردازش و دسترسی به داده های بیشتر، منجر به بازگشت هوش مصنوعی شد و این دوره نه تنها پایان عصر هوش مصنوعی بود، بلکه راه را برای الگوریتم های یادگیری ماشین و توسعه شبکه های عصبی هموار کرد. [؟، ؟، ؟]

## ۲.۱ انواع مدل یادگیری ماشین و شبکه های عصبی

### ۱.۲.۱ یادگیری ماشین: مروری کلی

یادگیری ماشین (Machine-Learning) شاخه ای از هوش مصنوعی است که به مدل های محاسباتی این امکان را می دهد الگو ها از داده ها را به طور خودکار یاد بگیرند و بتوانند تصمیم گیری کنند در واقع، هدف یادگیری ماشین این است که مدل ها بتوانند از داده ها الگو ها و روابط پنهان را استخراج کنند و به نتایج و رفتار و تصمیم های قابل اعتماد دست یابند.

### ۲.۲.۱ تقسیم بندی های اصلی در یادگیری ماشین

یادگیری ماشین به سه دسته اصلی تقسیم می شود: یادگیری با نظارت (Supervised-Learning) یادگیری بدون نظارت (Unsupervised-Learning) یادگیری تقویتی (Reinforcement-Learning)

### ۳.۲.۱ یادگیری نظارت شده (Supervised-Learning)

یادگیری نظارت شده یکی از رایج ترین روش ها در یادگیری ماشین شناخته می شود. که در آن از مجموعه داده های برچسب گذاری شده برای آموزش مدل استفاده می کنیم. هدف این الگوریتم تشخیص الگو ها در میان داده های ورودی است که این امکان را می دهد پیش بینی یا طبقه بندی هایی روی داده جدید انجام دهد. این نوع شامل دو الگوریتم Regression و classification می شود.

## طبقه‌بندی (Classification)

طبقه‌بندی یکی از مهم‌ترین و اصلی‌ترین وظایف در یادگیری نظارت شده است که هدف آن تخصیص داده‌ها به یک لیبل مشخص است. در این روش مدل با داده‌های برچسب دار (label) آموزش می‌بیند و یاد می‌گیرد که داده‌های جدید را بر اساس الگوها و ویژگی‌هایی که در داده‌های آموزشی دیده است، به دسته مناسب اختصاص دهد. از کاربرد های طبقه‌بندی می‌توان به تشخیص اسپم (spam)، تشخیص بیماری که آیا یک فرد مبتلا به بیماری هست یا نه، تشخیص چهره و ... استفاده کرد.

## رگرسیون (regression)

رگرسیون یکی از مهم‌ترین وظایف یادگیری ماشین است و هدف آن پیش‌بینی مقادیر پیوسته است. برخلاف طبقه‌بندی که خروجی آن دسته‌بندی مجزا است، در رگرسیون، خروجی یک مقدار پیوسته است و مدل یاد می‌گیرد روابط بین متغیرهای مستقل و متغیرهای هدف را شناسایی کند. از کاربرد های رگرسیون می‌توان به پیش‌بینی قیمت مسکن، پیش‌بینی آب و هوا و ... اشاره کرد.

## یادگیری تقویتی (Reinforcement-Learning)

یادگیری تقویتی نوعی یادگیری بر پایه پاداش و تنبیه است که در آن، مدل با محیط تعامل می‌کند و بر اساس پاداش یا تنبیه یاد می‌گیرد. برخلاف یادگیری نظارت‌شده و بدون نظارت، یادگیری تقویتی به مدل این امکان را می‌دهد، که از طریق آزمون و خطا بهترین راهکارها را برای انجام یک عمل یاد بگیرد. در این روش، مدل به جای برچسب، از یک تابع پاداش استفاده می‌کند که مشخص می‌کند چه اقداماتی باعث نتیجه بهینه می‌شود. از کاربرد های یادگیری تقویتی می‌توان به بازی‌ها (games)، کنترل رباتیک (robotic control)، سیستم‌های توصیه‌گر (Recommender-Systems) نام برد.

## ۴.۲.۱ معرفی چند مدل از الگوریتم یادگیری کلاسیک

نزدیک ترین همسایه (k-nearest-Neighbors)

KNN یکی از الگوریتم های ساده و کار آمد در یادگیری نظارت شده است که هم در دسته بندی و هم در رگرسیون کاربرد دارد این الگوریتم برای پیش بینی دسته بندی های یک نمونه جدید، به k نزدیک ترین داده ها در فضای ویژگی نگاه میکند و بر اساس اکثریت نزدیک ترین همسایه دسته بندی را انجام میدهد

مزایا:

- سادگی و قابل فهم بودن: این الگوریتم به سادگی با اندازه گیری فاصله بین نقاط داده کار می کند و بدون نیاز به آموزش مدل پیچیده قابل استفاده است.
- عملکرد خوب در داده های با تعداد ویژگی کم: در مسائلی که تعداد ویژگی ها کم است این الگوریتم به خوبی کار میکند.

معایب:

- حساسیت به داده های پرت: نقاط پرت می توانند به طور قابل توجهی بر نتایج تاثیر بگذارند.
- کندی در داده های بزرگ: این الگوریتم نیاز به محاسبه فاصله برای هر نقطه جدید دارد که در داده های بزرگ بار محاسباتی سنگین می شود.
- عدم کارایی در داده های با ابعاد بالا: در داده های با تعداد ویژگی های زیاد، کارایی الگوریتم کاهش می یابد.

### ۵.۲.۱ ماشین بردار پشتیبان: (Support-Vector-Machine)

الگوریتمی است که با یافتن یک ابر صفحه بهینه، داده ها را به کلاس مختلف تقسیم میکند. این الگوریتم یک ابر صفحه به دست می آورد که هدف آن حداکثر کردن فاصله میان داده های دو کلاس است و به این ترتیب میتواند طبقه بندی دقیقی داشته باشد.

مزایا:

- توانایی مقابله با داده های پیچیده و ابعاد بالا: SVM می تواند به خوبی با داده های چندبعدی و پیچیده کار کند.
- مقاومت در برابر بیش برآش (Overfitting): با استفاده از هسته ها (kernels) می توان داده های غیرخطی را نیز به فضای بالاتر برد و جداسازی بهتری انجام داد.

معایب:

- پیچیدگی محاسباتی: آموزش SVM به دلیل نیاز به حل مسائل بهینه سازی، در حجم های بالای داده محاسباتی زمان بر است.
- کارایی پایین در داده های پرت: در صورتی که داده ها شامل نقاط پرت زیادی باشند، دقت مدل کاهش می یابد.

### ۶.۲.۱ بیز ساده: (Naive-Bayes)

بیز ساده مبتنی بر قضیه بیز است و فرض میکند ویژگی ها به صورت شرطی مستقل از هم هستند. این مدل برای اولین بار در حوزه پردازش متن به کار رفت و هنوز هم در بسیاری از موارد مانند طبقه بندی ایمیل و تحلیل احساسات مورد استفاده قرار میگیرد. نایو بیز بر اساس احتمالات محاسبه میکند که یک نمونه جدید به کدام دسته تعلق دارد. این الگوریتم بر اساس قضیه بیز، احتمال تعلق



یک نمونه ب دسته را به ازای هر ویژگی محاسبه کرده و بیشترین احتمال را بر اساس جواب نهایی در نظر میگیرد.

مزایا:

- سرعت بالا: به دلیل محاسبات ساده و فرض استقلال ویژگی‌ها، Bayes Naive بسیار سریع و کم حجم است.
- کارایی در داده‌های کوچک: حتی با داده‌های کم، این الگوریتم عملکرد نسبتاً خوبی دارد.

### ۷.۲.۱ معایب:

- فرض استقلال ویژگی‌ها: فرض استقلال ویژگی‌ها ممکن است در بسیاری از مسائل واقعی صادق نباشد و این می‌تواند دقت مدل را کاهش دهد.
- حساسیت به داده‌های نادرست: در صورت داده‌های نادرست یا پرت، مدل ممکن است دقت کمتری داشته باشد.

### ۸.۲.۱ شبکه‌های عصبی بازگشتی (RNN) و شبکه‌های حافظه بلند مدت کوتاه مدت (LSTM)

شبکه عصبی بازگشتی با RNN ها و مدل هایی با حافظه بلند مدت تر مانند Lstm با هدف داده های ترتیبی و وابسته به زمان توسعه یافتند. این مدل ها به ویژه در تحلیل زبان طبیعی، صوت و پیش بینی سری های زمانی بسیار موفق عمل کرده اند. زیرا قادر به حفظ اطلاعات گذشته بودند و از این اطلاعات گذشته برای پیش بینی در لحظه حال و آینده استفاده میکنند.

## ۹.۲.۱ RNN:

مدل های اولیه شبکه های عصبی، مانند شبکه های چند لایه (Mlp) قادر به پردازش داده های مستقل و ثابت بودند و نمیتوانستند وابستگی های زمانی را یاد بگیرند. در بسیاری از مباحث دنیای واقعی، مانند تحلیل متن و صدا، داده ها به ترتیب خاصی وابسته هستند. به همین دلیل شبکه های Rnn معرفی شدند تا از اطلاعات پیشین در پردازش داده های بعدی استفاده کنند.

## RNN: ساختار و عملکرد

شبکه های Rnn دارای حلقه بازگشتی هستند که به مدل این امکان را می دهد اطلاعات را در توالی نگه دارند. و در هر گام زمانی ورودی فعلی  $x_t$  و وضعیت قبلی  $h_{t-1}$  به عنوان ورودی به نرون داده می شود.

$$h_t = \sigma(W \cdot x_t + U \cdot h_{t-1} + b) \quad (۱.۲.۱)$$

در اینجا:

$h_t$  وضعیت مخفی یا حالت در گام زمانی  $t$  است.

$W$  وزن هایی است که به ورودی  $x_t$  اعمال می شود.

$U$  وزن های اعمال شده بر وضعیت قبلی  $h_{t-1}$  است.

$b$  بایاس مدل است.

$\sigma$  تابع فعال سازی، معمولاً تانژانت هیپربولیک یا سیگموید.

با استفاده از این فرایند، مدل این توانایی را دارد که اطلاعات گذشته را در خود ذخیره کرده و در پردازش های بعدی آن ها را به کار بگیرد.

## ۱۰.۲.۱ مزایا و معایب Rnn:

در این قسمت به مزایا و معایب RNN میپردازیم.

مزایا:

- حفظ وابستگی زمانی: RNN قادر به پردازش توالی‌های طولانی است و می‌تواند اطلاعات را در طول توالی به خاطر بسپارد.
- کاربردهای گسترده در داده‌های ترتیبی: این مدل در تحلیل زبان طبیعی، پیش‌بینی سری‌های زمانی و پردازش صوتی بسیار موفق عمل می‌کند.

معایب:

- مشکل ناپدید شدن و انفجار گرادیان (Vanishing and Exploding Gradient): در فرآیند آموزش با روش پسانتشار، اگر توالی داده طولانی باشد، گرادیان‌ها ممکن است به سرعت کوچک یا بزرگ شوند، که منجر به ناپایداری آموزش و کاهش دقت می‌شود.
- محدودیت در پردازش توالی‌های بسیار بلند: RNN در حفظ اطلاعات طولانی مدت دچار مشکل است و برای پردازش وابستگی‌های طولانی، عملکرد ضعیفی دارد.

### ۱۱.۲.۱ شبکه‌های حافظه بلندمدت - کوتاه مدت (LSTM)

علل پیدایش lstm:

شبکه‌های lstm به عنوان یک راه حل برای یکی از بزرگترین مشکلات شبکه‌های عصبی بازگشتی (RNN) ها معرفی شدند. یکی از بزرگترین مشکلاتی که در شبکه‌های بازگشتی وجود داشت مشکل ناپدید شدن گرادیان (Vanishing Gradient) بود. این مشکل باعث میشد RNN ها قادر به یادگیری وابستگی‌های بلند مدت نباشند. برای درک عمیق تر ابتدا به توضیح مشکل ناپدید شدن گرادیان و سپس حل آن توسط lstm می‌پردازیم.

## Vanishing: Gradient

شبکه‌های RNN برای پردازش داده‌های ترتیبی، از حالت‌های بازگشتی استفاده می‌کنند. در فرآیند آموزش، RNN از الگوریتم پس‌انتشار خطا از طریق زمان (Backpropagation Through Time - BPTT) استفاده می‌شود. این الگوریتم گرادیان‌ها را برای به‌روزرسانی وزن‌ها محاسبه می‌کند. با این حال، به دلایل زیر، RNN‌ها وابستگی‌های بلندمدت ناکام می‌مانند:

- ضریب‌های بازگشتی کوچک‌تر از ۱: در فرآیند محاسبه گرادیان‌ها، اگر مقدار مشتقات یا ضرایب در هر مرحله کوچک‌تر از ۱ باشد، ضرب مکرر این مقادیر در طول توالی منجر به کوچک شدن گرادیان‌ها به سمت صفر می‌شود. این پدیده، ناپدید شدن گرادیان نام دارد.

فرمول کلی گرادیان در زمان  $t$  به صورت زیر است:

$$\frac{\partial L}{\partial W} = \prod_{k=1}^t \frac{\partial h_k}{\partial h_{k-1}} \cdot \frac{\partial h_t}{\partial L}$$

در اینجا،  $\frac{\partial h_k}{\partial h_{k-1}}$  می‌تواند مقداری کوچک‌تر از ۱ باشد، و ضرب مکرر این مشتقات باعث کاهش شدید مقدار گرادیان می‌شود.

- تاثیر مستقیم بر وزن‌ها: زمانی که گرادیان‌ها نزدیک به صفر شوند، وزن‌های مدل به‌طور موثری به‌روزرسانی نمی‌شوند. این امر مانع از یادگیری وابستگی‌های طولانی مدت در داده‌ها می‌شود.

## ۱۲.۲.۱ Lstm: ظهور

در سال ۱۹۹۷، Hochreiter Sepp و Schmidhuber Jürgen شبکه‌های حافظه بلندمدت - کوتاه‌مدت (LSTM) را معرفی کردند. انگیزه اصلی توسعه LSTM مشکل ناپدید شدن گرادیان در شبکه‌های RNN بود. این مشکل در مسائل یادگیری داده‌های ترتیبی طولانی باعث می‌شد RNN نتواند وابستگی‌های بلندمدت را به درستی یاد بگیرد.

راه حل LSTM برای پایداری جریان گرادیان‌ها:

LSTM با معرفی یک معماری جدید در شبکه‌های بازگشتی، جریان گرادیان‌ها را در طول توالی پایدار نگه می‌دارد. این کار از طریق اضافه کردن وضعیت سلولی (Cell) (State) و دروازه‌ها (Gates) به ساختار RNN انجام می‌شود. این اجزا به LSTM امکان می‌دهند که:

۱. اطلاعات غیرضروری را فراموش کند.

۲. اطلاعات مهم جدید را اضافه کند.

۳. اطلاعات مهم قبلی را حفظ کند.

### ۳.۱ اختار LSTM: نوآوری در مقایسه با RNN

LSTM شامل اجزای جدیدی است که به آن امکان مدیریت بهتر اطلاعات را می‌دهد:

#### ۱.۳.۱ وضعیت سلولی (Cell): (State)

مسیر اصلی ذخیره اطلاعات در LSTM است که می‌تواند اطلاعات مهم را در طول توالی حفظ کند. برخلاف RNN که وابسته به خروجی‌های بازگشتی  $h_t$  است، LSTM یک مسیر جداگانه برای عبور اطلاعات از وضعیت سلولی دارد که به حفظ گرادیان‌ها کمک می‌کند.

#### ۲.۳.۱ دروازه‌ها: (Gates)

دروازه‌ها نقش فیلترهای اطلاعاتی را دارند که جریان اطلاعات را در طول فرآیند یادگیری کنترل می‌کنند:

- دروازه فراموشی (Forget Gate): تعیین می‌کند چه اطلاعاتی از وضعیت سلولی باید حذف شود.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

مقدار فراموشی برای هر عنصر از وضعیت سلولی:  $f_t$

تابع سیگموئید که خروجی آن بین ۰ و ۱ است:  $\sigma$

وقتی  $f_t = 0$ ، اطلاعات به طور کامل حذف می‌شود؛ وقتی  $f_t = 1$ ، اطلاعات حفظ می‌شود.

• دروازه ورودی (Input Gate): تعیین می‌کند چه اطلاعات جدیدی باید به وضعیت سلولی اضافه شود.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

میزان اطلاعات جدیدی که به وضعیت سلولی وارد می‌شود:  $i_t$

مقدار جدید محاسبه شده برای اضافه شدن به وضعیت سلولی:  $\tilde{C}_t$

• دروازه خروجی (Output Gate): تعیین می‌کند چه اطلاعاتی از وضعیت سلولی به خروجی منتقل شود.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t \cdot \tanh(C_t)$$

### ۳.۳.۱ به‌روزرسانی وضعیت سلولی:

وضعیت سلولی  $C_t$  با استفاده از اطلاعات جدید و قدیمی به‌روزرسانی می‌شود:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$$

این ساختار باعث می‌شود که اطلاعات قدیمی مهم حفظ شده و اطلاعات غیرضروری حذف شوند.

### علت پایداری گرادیان در Lstm:

- حذف ضرب‌های مکرر: برخلاف RNN که به ضرب‌های مکرر وزن‌ها و گرادیان‌ها وابسته است، LSTM با مسیر جداگانه وضعیت سلولی، از کاهش نمایی گرادیان جلوگیری می‌کند.
- استفاده از توابع سیگموئید و تانژانت هیپربولیک: توابع سیگموئید در دروازه‌ها و تانژانت هیپربولیک در وضعیت سلولی باعث محدود کردن مقادیر و جلوگیری از انفجار گرادیان می‌شوند.
- مدیریت اطلاعات توسط دروازه‌ها: دروازه‌های فراموشی و ورودی به مدل اجازه می‌دهند تنها اطلاعات مهم حفظ شود و داده‌های غیرضروری حذف شوند، که این موضوع از پیچیدگی‌های محاسباتی غیرضروری جلوگیری می‌کند.

### ۴.۳.۱ مشکلات کلی rnn و lstm و ظهور ترانسفورمرها

شبکه‌های بازگشتی rnn ، lstm ها توانستند بسیاری از مشکلات و محدودیت های مدل های اولیه را حل کنند اما همچنان با چالش ها و محدودیت هایی مواجهه بودند که در مسائل پیچیده تر، مانند ترجمه زبان یا تحلیل داده های بلند مدت و حجیم، مشکلاتی را ایجاد میکردند. این مشکلات در نهایت باعث پیدایش ترانسفورمرها شد. که در اینجا مشکلات آن ها را بررسی میکنیم

#### مشکل وابستگی ترتیبی در RNN ها و LSTM ها

RNN ها و LSTM ها داده‌ها را به ترتیب پردازش می‌کنند، به این معنی که برای پردازش داده‌های گام زمانی  $t$ ، باید تمامی داده‌های قبلی  $(t - 1)$  را پردازش کرده باشند. این ویژگی مشکلات زیر را ایجاد می‌کند:

- غیرقابل موازی‌سازی: به دلیل وابستگی ترتیبی، پردازش داده‌ها به صورت موازی ممکن نیست، که باعث افزایش زمان محاسباتی می‌شود. این مشکل در داده‌های بلند، مانند متن‌های

طولانی یا سری‌های زمانی بزرگ، قابل توجه است.

- کندی آموزش و استنتاج: ترتیب خطی باعث می‌شود که زمان آموزش و پیش‌بینی مدل‌ها به شدت افزایش یابد، به ویژه زمانی که با حجم زیادی از داده‌ها سر و کار داریم.

محدودیت در یادگیری وابستگی‌های بسیار طولانی

با وجود پیشرفت LSTM در یادگیری وابستگی‌های بلندمدت نسبت به RNN این مدل‌ها همچنان در یادگیری وابستگی‌های بسیار بلند، مانند ارتباطات بین کلمات در دو جمله متفاوت یا درک ساختار کلی یک متن، محدودیت دارند:

- مشکل در داده‌های بسیار طولانی: حتی در LSTM ظرفیت حفظ اطلاعات محدود است و با افزایش طول توالی، دقت مدل کاهش می‌یابد.
- تاثیر تدریجی داده‌های اولیه: داده‌های ابتدایی توالی ممکن است با گذشت زمان اهمیت خود را از دست بدهند، زیرا گرادین‌ها به تدریج ضعیف‌تر می‌شوند.

پیچیدگی محاسباتی و حافظه

ها LSTM به دلیل ساختار پیچیده‌ای که شامل چندین ماتریس ضرب (برای دروازه‌های فراموشی، ورودی و خروجی) و به‌روزرسانی وضعیت سلول است، نیاز به حافظه و محاسبات زیادی دارند:

- نیاز به حافظه بیشتر: برای ذخیره وضعیت سلولی و گرادین‌ها، به حافظه بیشتری نسبت به مدل‌های ساده‌تر نیاز است.

- هزینه محاسباتی بالا: به خصوص در داده‌های بزرگ، محاسبات سنگین باعث کندی اجرای مدل‌ها می‌شود.



مشکل پردازش وابستگی‌های غیرمتوالی

ها RNN و LSTM به طور طبیعی برای یادگیری وابستگی‌های محلی و متوالی مناسب هستند. با این حال، در مسائل پیچیده مانند ترجمه زبان یا تحلیل متون، وابستگی‌های غیرمحلی و غیرمتوالی نیز وجود دارند. به عنوان مثال:

در یک جمله طولانی، ممکن است کلمه‌ای در ابتدای جمله با کلمه‌ای در انتهای جمله ارتباط معنایی داشته باشد. ها RNN و LSTM برای یادگیری این نوع وابستگی‌ها به شدت محدود هستند.

گرادیان‌های ناپایدار و مشکلات بهینه‌سازی:

با وجود بهبودهایی که LSTM نسبت به RNN در پایداری گرادیان‌ها ارائه داد، هنوز هم:

- مسائل گرادیان‌های ناپایدار: در توالی‌های بسیار بلند، گرادیان‌ها ممکن است همچنان کاهش یابند یا حتی در برخی موارد به حد انفجار برسند.

- مشکلات بهینه‌سازی: در مسائل پیچیده‌تر، یافتن مینیمم مناسب تابع هزینه با استفاده از ها RNN و LSTM دشوار است.

نیاز به مدلی با ظرفیت بیشتر و سرعت بالاتر:

- مدل‌های بزرگ‌تر: برای مسائل پیچیده‌تر، مدل‌هایی با تعداد پارامتر بیشتر نیاز است که RN-ها و LSTM به دلیل محدودیت در حافظه و پردازش، پاسخگوی این نیاز نیستند.

- کارایی در داده‌های چندوجهی: (Multimodal) برای داده‌هایی که شامل اطلاعات متنی، صوتی و تصویری هستند، ها RNN و LSTM توانایی لازم برای هم‌زمان پردازش این اطلاعات را ندارند.

وابستگی ترتیبی در RNN و LSTM یک مانع اساسی در برای استفاده از این مدل ها در مسئل پیچیده و بزرگ بود. که باعث به وجود آمدن ترانسفورمر ها شد. که با طراحی مبتنی بر موازی سازی و مکانیزم توجه،، این محدودیت را حذف کردند و راه حلی کارآمد تر برای پردازش داده های ترتیبی ارائه دادند.